Configuration Guide for BIG-IP® Application Security Manager™

version 10.2

MAN-0283-02

Product Version

This manual applies to product version 10.2 of the BIG-IP® Application Security Manager™.

Publication Date

This manual was published on July 2, 2010. Appendix B corrected on March 3, 2011. Chapter 6 corrected on November 29, 2011.

Legal Notices

Copyright

Copyright © 2011, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

F5, F5 Networks, the F5 logo, BIG-IP, 3-DNS, Access Policy Manager, APM, Acopia, Acopia Networks, Application Accelerator, Ask F5, Application Security Manager, ASM, ARX, Data Guard, Edge Client, Edge Gateway, Enterprise Manager, EM, FirePass, FreedomFabric, Global Traffic Manager, GTM, iControl, Intelligent Browser Referencing, Internet Control Architecture, IP Application Switch, iRules, Link Controller, LC, Local Traffic Manager, LTM, Message Security Module, MSM, NetCelera, OneConnect, Packet Velocity, Protocol Security Module, PSM, Secure Access Manager, SAM, SSL Accelerator, SYN Check, Traffic Management Operating System, TMOS, TrafficShield, Transparent Data Reduction, uRoam, VIPRION, WANJet, WAN Optimization Module, WOM, WebAccelerator, WA, and ZoneRunner are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by U.S. Patent 6,311,278. This list is believed to be current as of July 2, 2010.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD," "FreeBSD," "Mach" (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (http://www.apache.org/).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at http://www.perl.com.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (http://www.rrdtool.com/index.html) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (http://www.apache.org).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (http://www.nominum.com).

This product contains software developed by Broadcom Corporation, which is protected under the GNU General Public License.

This product includes the Zend Engine, freely available at http://www.zend.com.

This product contains software developed by NuSphere Corporation, which is protected under the GNU Lesser General Public License.

This product contains software developed by Erik Arvidsson and Emil A Eklund.

This product contains software developed by Aditus Consulting.

This product contains software developed by Dynarch.com, which is protected under the GNU Lesser General Public License, version 2.1 or above.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product contains software developed by InfoSoft Global (P) Limited.

This product includes software written by Steffen Beyer and licensed under the Perl Artistic License and the GPL.

This product includes software written by Makamaka Hannyaharamitu © 2007-2008.

Table of Contents

1
Introducing the Application Security Manager

Overview of the BIG-IP Application Security Manager 1-1
Summary of the Application Security Manager features 1-1
Configuration guide summary 1-2
Getting started with the user interface 1-3
Overview of components of the Configuration utility 1-3
Browser support for the Configuration utility 1-3
Finding help and technical support resources 1-4

2
Performing Essential Configuration Tasks

Overview of the essential configuration tasks 2-1
Defining a local traffic pool 2-2
Defining an application security class 2-3
Defining a local traffic virtual server 2-4
Running the Deployment wizard 2-5
Maintaining and monitoring the security policy 2-6

3
Working with Application Security Classes

What is an application security class? 3-1
Comparing application security classes and HTTP class profiles 3-1
Creating a basic application security class 3-2
Understanding the traffic classifiers 3-3
How the system applies the traffic classifiers 3-3
Classifying traffic using hosts 3-3
Classifying traffic using URI paths 3-5
Classifying traffic using headers 3-6
Classifying traffic using cookies 3-7
Configuring actions for the application security class 3-8
Rewriting a URI 3-9

4
Working with Web Applications

What is a web application? 4-1
Viewing the configured web applications 4-1
Configuring the properties of a web application 4-3
Configuring the web application language 4-3
Configuring the active security policy 4-4
Specifying the logging profile for a web application 4-4
Returning a web application to a new, unconfigured state 4-5
Working with web application groups 4-6
Creating a web application group 4-7
Removing a web application group 4-7
Working with a disabled web application 4-8
Viewing disabled web applications 4-8
Re-enabling a web application 4-8

5
Building a Security Policy Automatically

Overview of automatic policy building 5-1
Configuring automatic policy building 5-2
Configuring basic automatic policy building settings 5-2
Configuring advanced automatic policy building settings 5-4
Changing the policy type 5-6
Modifying security policy elements 5-9
Modifying automatic policy building options 5-11
Modifying automatic policy building rules 5-15
Modifying the list of trusted IP addresses 5-19
Restoring default values for automatic policy building 5-20
Viewing the automatic policy building status 5-21
Stopping and starting automatic policy building 5-23
Viewing automatic policy building logs 5-24

6
Manually Configuring Security Policies

Understanding security policies 6-1
Creating security policies 6-1
Configuring security policy properties 6-1
Configuring the security policy name and description 6-2
Viewing the web application associated with the security policy 6-2
Configuring the enforcement mode 6-3
Configuring the staging-tightening period 6-5
Enabling or disabling staging for attack signatures 6-6
Configuring the maximum HTTP header length 6-6
Configuring the maximum cookie header length 6-7
Configuring the allowed response status codes 6-8
Configuring dynamic session IDs in URLs 6-8
Activating iRule events 6-10
Configuring trusted XFF headers 6-11
Setting the active security policy for a web application 6-12
Determining when to set the active security policy 6-13
Validating HTTP protocol compliance 6-14
Understanding how HTTP protocol validation affects application security checks 6-14
Configuring HTTP protocol compliance validation 6-15
Adding file types 6-16
Creating allowed file types 6-17
Modifying file types 6-19
Removing file types 6-19
Disallowing specific file types 6-20
Configuring URLs 6-21
Creating an explicit URL 6-24
Removing a URL 6-25
Viewing or modifying the properties of a URL 6-25
Configuring URLs not allowed by the security policy 6-26
Configuring AMF security for URLs 6-27
Working with the URL character set 6-28
Configuring flows 6-30
Viewing the entire application flow 6-30
Viewing the flow to a URL 6-30
Adding a flow to a URL 6-31

Configuring a dynamic flow from a URL 6-32
Configuring login URLs to prevent forceful browsing 6-33
Masking sensitive data 6-35
Configuring allowed modified cookies 6-36
Editing an allowed modified cookie 6-37
Deleting an allowed modified cookie 6-38
Configuring mandatory headers 6-39
Configuring allowed methods 6-40
Configuring security policy blocking 6-41
Configuring the blocking policy 6-41
Configuring blocking properties for evasion techniques 6-44
Configuring blocking properties for HTTP protocol compliance 6-44
Configuring blocking properties for web services security 6-45
Configuring the response pages 6-46
Configuring CSRF protection 6-48

7
Configuring Anomaly Detection
What is anomaly detection? 7-1
Preventing DoS attacks for Layer 7 traffic 7-2
Recognizing DoS attacks 7-2
Configuring DoS attack mitigation 7-3
Mitigating brute force attacks 7-6
Configuring IP address enforcement 7-12
Detecting and preventing web scraping 7-13
Preventing web scraping detection on certain addresses 7-14

8
Maintaining Security Policies
Maintaining a security policy 8-1
Editing an existing security policy 8-2
Copying a security policy 8-3
Exporting a security policy 8-3
Importing a security policy 8-4
Merging two security policies 8-5
Removing a security policy from the configuration 8-6
Restoring a deleted security policy 8-7
Deleting a security policy permanently 8-7
Viewing and restoring an archived security policy 8-8
Reviewing a log of all security policy changes 8-9
Displaying security policies in a tree view 8-10
Using the security policy audit tools 8-11

9
Working with Wildcard Entities
Overview of wildcard entities 9-1
Understanding wildcard syntax 9-1
Understanding staging and tightening for wildcard entities 9-2
Understanding security policy enforcement for wildcard entities 9-4
Configuring wildcard file types 9-5
Creating wildcard file types 9-5
Modifying wildcard file types 9-6
Deleting wildcard file types 9-7

Sorting wildcard file types 9-8
Configuring wildcard URLs 9-9
Creating wildcard URLs 9-9
Modifying wildcard URLs 9-11
Deleting wildcard URLs 9-11
Sorting wildcard URLs 9-12
Configuring wildcard parameters 9-13
Creating wildcard parameters 9-13
Modifying wildcard parameters 9-15
Deleting wildcard parameters 9-15
Ordering wildcard parameters 9-16
Using wildcards for allowed modified cookie headers 9-18
Checking the status of wildcard tightening for allowed modified cookies 9-19
Enforcing allowed modified cookie wildcards 9-20

10
Working with Parameters

Understanding parameters 10-1
Understanding how the Security Enforcer processes parameters 10-1
Working with global parameters 10-2
Creating a global parameter 10-2
Editing the properties of a global parameter 10-4
Deleting a global parameter 10-4
Working with URL parameters 10-5
Creating a URL parameter 10-5
Editing the properties of a URL parameter 10-7
Deleting a URL parameter 10-7
Working with flow parameters 10-8
Creating a flow parameter 10-8
Editing the properties of a flow parameter 10-10
Deleting a flow parameter 10-11
Configuring parameter characteristics 10-12
Understanding parameter value types 10-12
Configuring static parameters 10-13
Configuring parameter characteristics for user-input parameters 10-14
Creating parameters without defined values 10-20
Allowing multiple occurrences of a parameter in a request 10-21
Making a flow parameter mandatory 10-22
Configuring XML parameters 10-23
Working with dynamic parameters and extractions 10-24
Configuring dynamic content value parameters 10-24
Viewing the list of extractions 10-27
Configuring parameter characteristics for dynamic parameter names 10-27
Working with the parameter character sets 10-29
Viewing and modifying the default parameter value character set 10-29
Viewing and modifying the default parameter name character set 10-30
Configuring sensitive parameters 10-31
Configuring navigation parameters 10-32

11
Working with Attack Signatures

Overview of attack signatures 11-1
Understanding the global attack signatures pool 11-1
Overview of attack signature sets 11-2
Understanding how the system uses attack signatures 11-2
Types of attacks that attack signatures detect 11-3
Managing the attack signatures pool 11-6
Working with the attack signatures pool filter 11-6
Viewing attack signature details 11-8
Updating the system-supplied attack signatures 11-10
Important considerations when updating attack signatures 11-10
Configuring automatic updates for system-supplied attack signatures 11-11
Configuring manual updates for system-supplied attack signatures 11-11
Viewing information about the most recent update 11-12
Receiving email notification of attack signature updates 11-12
Working with attack signature sets 11-13
Viewing system-supplied signature sets 11-13
Creating an attack signature set 11-14
Editing used-defined attack signature sets 11-16
Deleting a user-defined attack signature set 11-16
Assigning attack signature sets to a security policy 11-17
Viewing the attack signature sets for a specific security policy 11-17
Viewing all attack signatures for a security policy 11-18
Disabling an attack signature in a security policy 11-19
Modifying the blocking policy for an attack signature set 11-20
Understanding attack signature staging 11-21
Managing signatures in staging that generate learning suggestions 11-21
Enabling or disabling signatures in staging 11-23
Enforcing all attack signatures 11-24
Managing user-defined attack signatures 11-25
Creating a user-defined attack signature 11-26
Modifying a user-defined attack signature 11-27
Deleting a user-defined attack signature 11-27
Importing user-defined attack signatures 11-28
Exporting user-defined attack signatures 11-29

12
Protecting XML Applications

Getting started with XML security 12-1
Configuring security for SOAP web services 12-3
Implementing web services security 12-5
Uploading certificates 12-6
Enabling encryption, decryption, signing, and verification of SOAP messages 12-7
Managing SOAP methods 12-13
Configuring security for XML content 12-14
Fine-tuning XML defense configuration 12-16
Masking sensitive XML data 12-19
Associating an XML profile with a URL 12-20
Associating an XML profile with a parameter 12-22
Modifying XML security profiles 12-23
Editing an XML profile 12-23
Deleting an XML profile 12-24

13
Refining the Security Policy Using Learning

Overview of the learning process 13-1
Working with learning suggestions 13-2
Viewing all requests that trigger a specific learning suggestion 13-3
Viewing the details of a specific request 13-4
Viewing all requests for a specific web application 13-6
Accepting or clearing learning suggestions 13-7
Accepting a learning suggestion 13-7
Clearing a learning suggestion 13-8
Working with entities in staging or with tightening enabled 13-9
Understanding tightening 13-10
Understanding staging 13-11
Reviewing staging and tightening status 13-12
Adding new entities to the security policy from staging or tightening 13-13
Processing learning suggestions that require user interpretation 13-15
Disabling violations 13-16
Clearing violations 13-17
Viewing ignored entities 13-18
Removing items from the ignored entities list 13-18
Adding and deleting ignored IP addresses 13-19

14
Configuring General System Options

Overview of general system options 14-1
Configuring interface and system preferences 14-2
Configuring external anti-virus protection 14-3
Configuring user accounts for security policy editing 14-4
Configuring logging profiles for web application data 14-5
Creating a logging profile for local storage 14-5
Configuring a logging profile for remote storage 14-6
Configuring a logging profile for a reporting server 14-8
Configuring a logging profile if using ArcSight logs 14-9
Configuring the storage filter 14-10
Setting event severity levels for security policy violations 14-11
Viewing the application security logs 14-12
Validating regular expressions 14-13
Configuring an SMTP mail server 14-14

15
Displaying Reports

Overview of the reporting tools 15-1
Displaying an application security overview 15-2
Reviewing details about requests 15-4
Exporting requests 15-7
Clearing requests 15-7
Viewing charts 15-8
Interpreting graphical charts 15-10
Scheduling and sending graphical charts using email 15-11
Viewing anomaly statistics 15-12
Viewing DoS Attacks reports 15-12
Viewing Brute Force Attack reports 15-13
Viewing IP Enforcer statistics 15-13
Viewing web scraping statistics 15-14

Viewing PCI Compliance reports 15-15
Filtering reports 15-17
Monitoring CPU usage 15-18

A
Security Policy Violations

Introducing security policy violations A-1
Viewing descriptions of violations A-1
RFC violations A-3
Access violations A-5
Length violations A-6
Input violations A-8
Cookie violations A-11
Negative security violations A-12
Determining the type of attack detected by an attack signature A-13
Filtering requests by attack type A-13

B
Working with the Application-Ready Security Policies

Understanding application-ready security policies B-1
Using the Deployment wizard to implement application-ready security policies B-1
Using the Rapid Deployment security policy B-2
Overview of the Rapid Deployment security policy features B-2
Using the ActiveSync security policy B-3
Overview of the ActiveSync security policy features B-3
Configuring the system to secure the ActiveSync application B-3
Using the OWA Exchange 2003 security policy B-4
Overview of the OWA Exchange 2003 security policy features B-4
Configuring the system to secure the OWA 2003 application B-4
Using the OWA Exchange 2007 security policy B-5
Overview of the OWA Exchange 2007 security policy features B-5
Configuring the system to secure the OWA 2007 application B-5
Using the SharePoint 2003 security policy B-6
Overview of the SharePoint 2003 security policy features B-6
Configuring the system to secure the SharePoint 2003 application B-6
Using the SharePoint 2007 security policy B-7
Overview of the SharePoint 2007 security policy features B-7
Configuring the system to secure the SharePoint 2007 application B-7
Using the Lotus Domino 6.5 security policy B-8
Overview of the Lotus Domino 6.5 security policy features B-8
Configuring the system to protect the Lotus Domino 6.5 application B-8
Using the Oracle Applications 10g security policy B-9
Overview of the Oracle Applications 10g security policy features B-9
Configuring the system to protect the Oracle Applications 10g application B-9
Using the Oracle Applications 11i security policy B-10
Overview of the Oracle Applications 11i security policy features B-10
Configuring the system to protect the Oracle Applications 11i application B-10
Using the PeopleSoft Portal 9 security policy B-11
Overview of the PeopleSoft Portal 9 security policy features B-11
Configuring the system to protect the PeopleSoft Portal 9 application B-11
Using the SAP NetWeaver security policy B-12
Overview of the SAP NetWeaver security policy features B-12
Configuring the system to protect the SAP NetWeaver application B-12

Using the WhiteHat Sentinel Baseline security policy B-13
Overview of the WhiteHat Baseline security policy features B-13
Configuring the system to work with WhiteHat Sentinel B-13
Managing large file uploads when using the application-ready security policies B-14

C
Syntax for Creating User-Defined Attack Signatures
Writing rules for user-defined attack signatures C-1
Understanding the rule options C-1
Overview of rule option scopes C-3
Scope modifiers for the pcre rule option C-3
A note about normalization C-4
Syntax for attack signature rules C-5
Using the content rule option C-5
Using the uricontent rule option C-5
Using the headercontent rule option C-6
Using the valuecontent rule option C-6
Using the pcre rule option C-6
Using the reference rule option C-8
Using the nocase modifier C-8
Using the offset modifier C-9
Using the depth modifier C-9
Using the distance modifier C-10
Using the within modifier C-11
Using the objonly modifier C-12
Using the norm modifier C-12
Using character escaping C-13
Syntax considerations for parameter attack signatures C-14
Syntax considerations for response attack signatures C-14
Combining rule options C-14
Rule combination example C-15

D
Internal Parameters for Advanced Configuration
Overview of internal parameters D-1
Viewing internal parameters D-4
Restoring the default settings for internal parameters D-5

E
Upgrading HTTP Security Profiles to Security Policies
Overview of the Migration wizard E-1
Performing the migration E-2

F
Running Application Security Manager on the VIPRION Chassis
Overview of running Application Security Manager on the VIPRION chassis F-1
Viewing cluster statistics F-2
Viewing VIPRION cluster member synchronization status F-2

Glossary

Index

1

Introducing the Application Security Manager

- Overview of the BIG-IP Application Security Manager
- Getting started with the user interface
- Finding help and technical support resources

Overview of the BIG-IP Application Security Manager

The BIG-IP® Application Security Manager™ protects mission-critical enterprise Web infrastructure against application-layer attacks, and monitors the protected web applications. The Application Security Manager can prevent a variety of web application attacks, such as:

- Manipulation of cookies or hidden fields
- SQL injection attacks intended to expose confidential information or to corrupt content
- Malicious exploitations of the application memory buffer to stop services, to get shell access, and to propagate worms
- Unauthorized changes to server content using HTTP DELETE and PUT methods
- Attempts aimed at causing the web application to be unavailable or to respond slowly to legitimate users
- Layer 7 denial-of-service and brute force attacks
- Unknown threats, also known as zero-day threats

The system can automatically develop a security policy to protect against security threats, and you can configure additional protections and customize the system response to threats.

Summary of the Application Security Manager features

The Application Security Manager includes the following features.

- **Integrated platform guaranteeing the delivery of secure application traffic**

  Built on F5 Networks' TMOS® architecture, the ICSA-certified, positive-security Application Security Manager is fully integrated with the BIG-IP Local Traffic Manager™.

- **Automated security policy building**

  Application Security Manager uses an auto-adaptive approach to application delivery security, where the security policy is automatically built and updated based on observed traffic patterns. A Deployment wizard helps you create a security policy for your environment. Then the automated policy building feature, called the Real Traffic Policy Builder™, examines requests and responses, and populates the security policy with legitimate security policy elements, based on what it finds in the traffic.

- **Attack Signature protection**

  The Attack Signatures in the Application Security Manager provide protection from generalized and known application attacks such as worms, vulnerabilities, and requests for restricted files and URLs. The Attack Signatures Update feature provides current, up-to-date signatures, so that your applications are protected from new attacks and threats.

- **Positive security model**
  The Application Security Manager creates a robust positive security policy to completely protect web applications from targeted web application layer threats, such as buffer overflows, SQL injection, cross-site scripting, parameter tampering, cookie poisoning, and others, by allowing only valid application transactions. The positive security model is based on a combination of valid user session context and valid user input, as well as a valid application response.
- **Integrated, simplified management**
  The browser-based Configuration utility provides network device configuration, centralized visual security policy management, and easy-to-read audit reports. Additional tools provide a highly automated and visual security policy building mechanism, based on a proprietary Policy Builder that automatically builds a map of all the valid application transactions and drastically simplifies the security policy management.
- **Configurable security levels**
  The Application Security Manager offers varying levels of security, from general protection of web site elements such as file types and character sets, to tailored, highly granular, application-specific security policies. This flexibility provides enterprises the ability to choose the level of security they need, and reduce management costs based on the level of protection and risks acceptable in their business environment.
- **Role-based administration**
  The BIG-IP system supports role-based administration, which you can use to restrict access to various components of the product. For example, users with the **Web Application Security Editor** role can audit and maintain application security policies on a specific partition, but they have no access to general BIG-IP system administration.

Configuration guide summary

To use this guide, you must have installed the BIG-IP system, and have licensed and provisioned Application Security Manager. This guide focuses on configuring the application security components, including:

- Application security classes
- Web applications
- Security policies
- XML Profiles
- Policy Builder
- Monitoring, statistics, and logging

This configuration guide also contains basic information on configuring a local traffic virtual server to use an application security class to protect the web application resources. The ***application security class*** is the bridge between the local traffic components and the application security

components. For detailed information on configuring the local traffic objects, refer to the ***Configuration Guide for BIG-IP® Local Traffic Manager™***.

When you provision Application Security Manager, the Protocol Security Module™ is also included on the system and available for use (without needing to be provisioned separately). For information on working with protocol security objects, refer to the ***Configuration Guide for BIG-IP® Protocol Security Module™***.

Getting started with the user interface

The browser-based graphical user interface for the BIG-IP system is called the Configuration utility. You log on and use the Configuration utility to set up the system and configure the Application Security Manager.

Overview of components of the Configuration utility

The Configuration utility contains the following components:

- **The identification and messages area**
  The identification and messages area of the Configuration utility is the screen region that is above the navigation pane, the menu bar, and the body. In this area, you find the system identification, including the host name and management IP address. This area is also where certain system messages display, for example **Activation Successful**, which appears after a successful licensing process.
- **The navigation pane**
  The navigation pane, on the left side of the screen, contains the Main tab, the Help tab, and the About tab. The Main tab provides links to the major configuration objects. The Help tab provides context-sensitive help for each screen in the Configuration utility. The About tab provides overview information about the BIG-IP system.
- **The menu bar**
  The menu bar, which is below the identification and messages area, and above the body, provides links to additional screens.
- **The body**
  The body is the screen area where the configuration settings display, and where the user configures the system.

Browser support for the Configuration utility

For a list of the supported browsers for the Configuration utility, refer to the current release notes on the Ask F5℠ Knowledge Base web site, **https://support.f5.com**.

Finding help and technical support resources

You can find additional technical documentation and product information using the following resources:

- **Online help for Application Security components**
  The Configuration utility has online help for each screen. The online help contains descriptions of each control and setting on the screen. Click the Help tab in the left navigation pane to view the online help.
- **Welcome screen in the Configuration utility**
  The Welcome screen in the Configuration utility contains links to many useful web sites and resources, including the Ask F5SM Knowledge Base, the F5 Solution Center, the F5 DevCentral web site, plug-ins, SNMP MIBs, and SSH clients. The Welcome screen is shown previously in Figure , on page 1-3.
- **F5 Networks Technical Support web site**
  The F5 Networks Technical Support web site, **https://support.f5.com**, provides the latest documentation for the product, including:
  - Release notes for the Application Security Manager, Local Traffic Manager, and Protocol Security module
  - ***BIG-IP® Application Security Manager™: Getting Started Guide***
  - ***Configuration Guide for BIG-IP® Local Traffic Manager™***
  - ***Configuration Guide for BIG-IP® Protocol Security Module™***
  - ***BIG-IP® Systems: Getting Started Guide***
  - ***TMOS® Management Guide for BIG-IP® Systems***
  - Technical notes
  - AskF5SM Knowledge Base

To access this site, you need to register at **https://support.f5.com**.

2

Performing Essential Configuration Tasks

- Overview of the essential configuration tasks
- Defining a local traffic pool
- Defining an application security class
- Defining a local traffic virtual server
- Running the Deployment wizard
- Maintaining and monitoring the security policy

Overview of the essential configuration tasks

This chapter is your guide to the essential configuration tasks you must complete to initially create and refine a standard security policy for a web application on the Application Security Manager™. Implementing a security policy for a web application has two phases: setting up the local traffic network, and creating the application security configuration.

These are the phase one configuration tasks:

- **Define a local traffic pool.**
  The local traffic pool contains the web server or application server resources that host the web application that you want to protect with a security policy. You create the local traffic pool, and then associate the pool with an application security class. See *Defining a local traffic pool*, on page 2-2, for more information.
- **Define an application security class.**
  When you define an application security class (with application security enabled), the system automatically creates a corresponding web application in the Application Security Manager. See *Defining an application security class*, on page 2-3, for more information.
- **Define a local traffic virtual server that uses the application security class as a resource.**
  The local traffic virtual server load balances the network resources that host the web application you are securing. The application security class is the bridge that links the security policy to the web application traffic through the virtual server. You configure the virtual server, and then associate the application security class with the virtual server. See *Defining a local traffic virtual server*, on page 2-4, for more information.

These are the phase two configuration tasks:

- **Run the Deployment wizard.**
  Using the Deployment wizard, you create a security policy, based on one of several typical deployment scenarios. See *Running the Deployment wizard*, on page 2-5, for more information.
- **Periodically review the security policy settings.**
  To ensure that the security policy is providing adequate application security, review the requests, monitoring, and statistics information on a regular basis. See *Maintaining and monitoring the security policy*, on page 2-6, for more information.

This chapter describes the general tasks that you perform to configure a security policy for a web application hosted on a local traffic virtual server. The chapter does not address specific deployments or environments. For additional implementations that address the needs of a particular

environment, refer to the ***BIG-IP® Application Security Manager™: Getting Started Guide***, which is available in the Ask F5SM Knowledge Base, **https://support.f5.com**.

Important

The tasks described in this chapter begin after you have installed the BIG-IP system, and have licensed and provisioned the Application Security Manager. If you have not yet completed these activities, refer to the ***BIG-IP® Systems: Getting Started Guide****, and the* ***TMOS® Management Guide for BIG-IP® Systems*** *for additional information.*

Defining a local traffic pool

The first essential configuration task is to define a local traffic pool. The local traffic pool contains the resources that host the actual web application content that you want to protect with the security policy.

The following procedure outlines only the basic pool configuration. For detailed information on configuring pools, refer to the ***Configuration Guide for BIG-IP® Local Traffic Manager™***, which is available in the Ask F5SM Knowledge Base, **https://support.f5.com**.

To define a local traffic pool

1. In the navigation pane, expand **Local Traffic**, and then click **Pools**.
   The Pool List screen opens.
2. Click the **Create** button.
   The New Pool screen opens.
3. In the Configuration area, in the **Name** box, type a name for the pool.
4. In the Resources area, for the **New Members** setting, in the **Address** box, type the IP address for the web server or application server that hosts the web application.
5. In the **Service Por**t box, type the service port number (for example, type **80** for the HTTP service), or select a service name from the list.
6. Click the **Add** button to add the resource to the **New Members** list.
7. Click the **Finished** button.
   The screen refreshes and the system displays the new pool in the pools list.

Defining an application security class

The second essential configuration task is to configure an application security class. An ***application security class*** is the logical bridge, or link, between the local traffic components and the application security components of the BIG-IP system. You use the application security class to specify to which incoming HTTP traffic the system applies application security before the virtual server forwards the traffic to the web application.

When you configure an application security class, the system automatically creates a default web application in the Application Security Manager. For more information on application security classes, see Chapter 3, *Working with Application Security Classes*. For information about web applications, see Chapter 4, *Working with Web Applications*.

To create an application security class

1. In the navigation pane, expand **Application Security** and then click **Classes**.
   The HTTP Class Profiles screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. In the General Properties area, in the **Name** box, type a name for the application security class.
4. In the Configuration area, leave all of the settings at the defaults.
5. In the Actions area, check the **Custom** box located at the right of the **Send To** setting, and select **Pool** from the list.
   The screen refreshes, and you see additional settings.
6. For the **Pool** setting, select the local traffic pool that you created.
7. Click **Finished**.
   The system adds the class, a default web application (which is unconfigured at this point), and displays the HTTP Class Profiles screen.

◆ Note

*In the Configuration utility, the application security class and the HTTP Class Profile are different labels for the same object. The difference between the two objects is that, for the application security class, the **Application Security** setting is enabled by default. If you disable the **Application Security** setting on an application security class, you effectively turn off application security for the associated web application.*

Defining a local traffic virtual server

The next essential configuration task is to define a virtual server on the local area network. The virtual server processes the incoming traffic, which includes applying the application security class to incoming HTTP traffic.

The following procedure outlines only the basic virtual server configuration. For detailed information on virtual servers, including SSL virtual servers, and other local traffic components, refer to the ***Configuration Guide for BIG-IP® Local Traffic Manager***™, which is available in the Ask F5℠ Knowledge Base, **https://support.f5.com**.

To configure a virtual server

1. In the navigation pane, expand **Local Traffic**, and then click **Virtual Servers**.
   The Virtual Server List screen opens.
2. Click the **Create** button.
   The New Virtual Server screen opens.
3. In the **Name** box, type a name for the virtual server.
4. In the **Destination** option, select **Host**, and type an IP address.
5. In the **Service Port** box, type **80**. Alternately, you can select **HTTP** from the list.
6. In the Configuration section, from the **HTTP Profile** list, select **http**.
7. In the Resources section, for the **HTTP Class Profiles** setting, from the **Available** list, select the application security class that you created, and click the Move button (<<) to add the class to the **Enabled** list.
8. Click **Finished**.
   The system updates the configuration, and the Virtual Server List screen opens, where you can see your newly created virtual server.

◆ Note

For virtual servers that load balance resources for a web application that is protected by the Application Security Manager, you must configure an HTTP profile in addition to the application security class. Refer to steps 6 and 7 in the previous procedure.

Running the Deployment wizard

After you have completed the phase one tasks, which set up the local area network, you are ready for the phase two tasks. The phase two tasks include configuring the security policy, and monitoring the security policy.

You build a security policy for a new web application using the Deployment wizard. The Deployment wizard automates the fundamental tasks required to initially build and deploy a security policy. The Deployment wizard provides several ***deployment scenarios***, which represent several typical environments that use application security, to guide you through the configuration process.

To start the Deployment wizard

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Active Security Policy column of the web application for which you want to create a security policy, click the **Configure Security Policy** link.
   The Select Deployment Scenario screen opens.
3. For the **Deployment Scenario** setting, select the appropriate environment:
   - **Production Site (Untrusted Traffic)**
     Select this option if the system is in a production area.
   - **QA Lab (Trusted Traffic)**
     Select this option if the system is in a test area within the corporate network.
   - **Web Services (XML + WSDL/User Schema)**
     Select this option to protect a web service or XML application.
   - **Manual Deployment**
     Select this option for rapid deployment or to create a security policy from a security policy template.
4. Follow through the screens of the wizard.
   The **Description** area of each wizard screen provides additional information about the screen. The online help describes each of the options on the screen.

For more information about running the Deployment wizard for a specific deployment scenario, refer to the ***BIG-IP® Application Security Manager™: Getting Started Guide.***

Maintaining and monitoring the security policy

The Application Security Manager provides many reporting and monitoring tools, so that you can view and analyze the violations that the system detects in the traffic passing through the web application. By actively using the reporting and monitoring tools, you can be assured that your web applications are fully protected.

To view the monitoring tools

1. In the navigation pane, expand **Application Security**, and click **Reporting**.
   The Requests screen opens.
2. Using the menu bar, choose the report that you want to view.
3. On each screen, you can use the **Filter** option to customize and refine the reports.

For additional information and details about the reporting tools, refer to Chapter 15, *Displaying Reports*.

3

Working with Application Security Classes

- What is an application security class?
- Understanding the traffic classifiers
- Configuring actions for the application security class

What is an application security class?

An ***application security class*** is the logical bridge, or link, between the local traffic components and the application security components. You create one or more application security classes, and then assign them as resources for one or more local traffic virtual servers. When the virtual server receives an HTTP request, it applies the application security classes, in the listed order, and if the traffic classifiers find a match in the request, the system routes the request to the Application Security Manager™.

In the application security class, the ***traffic classifiers*** specify which incoming HTTP traffic should be routed through the Application Security Manager. The traffic classifiers use different elements of an HTTP request, including host header values, URI paths, other headers and values, and cookie names (or a combination of all of these), to determine which requests go to the Application Security Manager. For requests that match the traffic classifiers, the Application Security Manager applies the active security policy to the designated traffic, and processes the traffic according to the security policy settings.

When you configure an application security class (or an HTTP class profile with application security enabled), the system automatically creates a default web application in Application Security Manager. You then configure an active security policy for the web application. For complex applications, you can create more than one application security class if you need to apply different security policies to different parts of the application.

Comparing application security classes and HTTP class profiles

The application security class and the HTTP class profile are two names for the same basic object in the Configuration utility. The primary difference between the two objects is that when you create an application security class, the system automatically enables the **Application Security** setting. HTTP class profiles exist (without the **Application Security** option enabled) on every Local Traffic Manager system, and they are used to classify HTTP traffic. By default, when you create an HTTP class profile, the **Application Security** setting is not enabled (and the setting is available only if you have Application Security Manager provisioned on the system).

You create application security classes from the **Application Security** section of the navigation pane. You create HTTP class profiles from the **Profiles** option in the **Local Traffic** section of the Main tab. (For information on the generic HTTP class profile, see the *Managing Protocol Profile* chapter, in the ***Configuration Guide for BIG-IP® Local Traffic Manager***™.)

◆ Tip

*F5 Networks recommends that you create the application security classes from the **Application Security** section on the Main tab of the navigation pane so that the system automatically enables the application security option for you.*

Creating a basic application security class

A basic application security class simply routes all HTTP traffic through the Application Security Manager.

To create a basic application security class

1. On the Main tab of the navigation pane, expand **Application Security** and then click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
   Note that in the application security configuration, the corresponding web application and security policy also use this name.
4. Leave all of the traffic classifier settings at the default, which is **Match All**.
5. Above and on the right of the Actions area, select the **Custom** check box to enable Actions options.
6. For the **Send To** setting, select **Pool** from the list.
   The screen refreshes, and the action settings are all enabled.
7. For the **Pool** setting, select the local traffic pool that contains the web server resources for your web application.

   ***Note**: If you have not already configured a local traffic pool, refer to Defining a local traffic pool, on page 2-2.*
8. Click **Finished**.
   The system adds the new application security class, and also automatically creates a web application with the same name, and creates a security policy with the same name with a **_default** suffix.

Tip

For additional information about BIG-IP HTTP class traffic flow, see Solution 8018 in the Ask F5^{SM} Knowledge Base, ***https://support.f5.com/kb/en-us/solutions/public/8000/000/sol8018.html****.*

Understanding the traffic classifiers

You can use the traffic classifiers in the application security class to specify exactly which traffic goes through the Application Security Manager before it reaches the web application resources. The traffic classifiers perform pattern matching against HTTP requests, based either on wildcard strings or on regular expressions. When the traffic classifier finds a match in an HTTP request, the system forwards that request to the Application Security Manager. The Application Security Manager then applies the active security policy to the request.

The traffic classifiers perform pattern matching using either literal strings or regular expressions. The literal strings can include wildcard characters, such as asterisk (*) or question mark (?). The regular expressions use the Tcl regular expression syntax. You can use a mixture of matching types within each traffic classifier.

◆ Note

Pattern-matching traffic classifiers are case-sensitive; that is, ***www.F5.com*** *is not the same as* ***www.f5.com****. See the F5 Dev Central web site,* ***http://devcentral.f5.com****, for information on Tcl expressions and syntax.*

How the system applies the traffic classifiers

You can configure one or more traffic classifiers in each application security class. If the traffic classifier has multiple matching objects within its list, the system looks for a match until it finds one, and forwards the request when it does. If you configure more than one type of classifier (for example, you configure both a URI path and a header traffic classifier), the system performs the pattern matching and forwards to the Application Security Manager only the traffic that matches both traffic classifier types. If you configure multiple entries within each traffic classifier list, the system performs the pattern matching until it finds a match.

Classifying traffic using hosts

You can use the Hosts traffic classifier to specify hosts whose traffic you want to direct through the Application Security Manager. When you use the Hosts traffic classifier, the system performs pattern matching against the information contained in the **Host** header in a request.

◆ Tip

Just by configuring the valid host headers for the web application, you acquire immunity to most of the worms that are spread by an IP address as a value in the Host header.

To configure an application security class using the Hosts traffic classifier

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
4. For the **Configuration** setting, select the **Custom** check box to enable the Configuration options.
5. For the **Hosts** setting, select **Match only**.
   The screen refreshes, and you see the **Host List** setting.
6. Add hosts to the **Host List** as needed:
   a) In the **Host** box, type the name of the host for which the system routes HTTP traffic through the Application Security Manager.
   b) For **Entry Type**, select **Pattern String** or **Regular Expression (regex)**.
   c) Click **Add**.
      The host is added to the list.
7. Configure the remaining settings as needed.
8. Click **Finished**.
   The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

Tip

For information on the other options on this screen, click the Help tab in the navigation pane.

Classifying traffic using URI paths

You can use the URI Paths traffic classifier to specify one or more URI paths whose requests you want to direct through the Application Security Manager. When you use the URI Paths traffic classifier, the system performs pattern matching against the URI path in a request.

To configure an application security class using the URI Paths traffic classifier

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
4. For the **Configuration** setting, select the **Custom** check box to enable the Configuration options.
5. For the **URI Paths** setting, select **Match only**.
   The screen refreshes, and you see the **URI Path List** setting.
6. Add URIs to the **URI Path List** as needed.
   a) In the **URI Path** box, type the URI path for which the system routes HTTP traffic through the Application Security Manager.
   b) For **Entry Type**, select **Pattern String** or **Regular Expression (regex)**.
   c) Click **Add**.
   The URI is added to the list.
7. Configure the remaining settings as needed.
8. Click **Finished**.
   The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

◆ Tip

For information on the other options on this screen, click the Help tab in the navigation pane.

Classifying traffic using headers

You can use the Headers traffic classifier to specify one or more headers whose associated requests you want to direct through the Application Security Manager. When you use the Headers traffic classifier, the system performs pattern matching against the headers and their values in a request.

Note

If you want to classify traffic using the Cookie header, use the Cookies traffic classifier instead of the Headers traffic classifier. See ***Classifying traffic using cookies****, on page 3-7, for more information.*

To configure an application security class using the Headers traffic classifier

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
4. Above and on the right of the Configuration area, select the **Custom** check box to enable the Configuration options.
5. For the **Headers** setting, select **Match Only**.
   The screen refreshes, and you see the **Header List** setting.
6. Add headers and their values to the **Header List** as needed:
   a) In the **Header** box, type the header. Include the colon when you add headers to this list, for example: **User-Agent:<value>**.
   b) For **Entry Type**, select **Pattern String** or **Regular Expression (regex)**.
   When you select **Regular Expression (regex)**, the system prepends **(regex)** when you add the object to the list.
   c) Click **Add**.
   The header is added to the list.
7. Configure the remaining settings as needed.
8. Click **Finished**.
   The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

Tip

For information on the other options on this screen, click the Help tab in the navigation pane.

Classifying traffic using cookies

You can use the Cookies traffic classifier to specify one or more cookies whose associated requests you want to direct through the Application Security Manager. When you use the Cookies traffic classifier, the system performs pattern matching against the cookie name information in the **Cookie** header in a request.

To configure an application security class using the Cookies traffic classifier

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
4. For the **Configuration** setting, select the **Custom** check box to enable the Configuration options.
5. For the **Cookies** setting, select **Match Only**.
   The screen refreshes, and you see the **Cookie List** setting.
6. Add cookie names to the **Cookie List** as needed:
   a) In the **Cookie** box, type the cookie data.
   b) For **Entry Type**, select **Pattern String** or **Regular Expression (regex)**.
   c) Click **Add**.
      The cookie is added to the list.
7. Configure the remaining settings as needed.
8. Click **Finished**.
   The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

◆ Tip

For information on the other options on this screen, click the Help tab in the navigation pane.

Configuring actions for the application security class

The actions of the application security class designate what the system does with the traffic when the traffic matches one or more of the traffic classifier criteria. The actions for the application security class are as follows.

- **None**
  When you use the none action, the system does nothing with the traffic within the context of this application security class. The system may process the request according to other settings for the virtual server, for example, forward the request to the virtual server's default pool.
- **Send to pool**
  When you use the send to pool action, the system sends the traffic to the local traffic pool specified in the **Pool** setting. In this case, traffic is not sent to the Application Security Manager, nor to the pool specified in the virtual server (unless it is the same pool).
- **Redirect to another resource**
  When you use the redirect action, the system sends any traffic that matches (based on the full HTTP URI) to another resource on the network. You can use Tcl expressions to create a custom redirection. See the F5 Dev Central web site, **http://devcentral.f5.com**, for information on Tcl expressions and syntax.

To configure an action for the application security class

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a unique name for the application security class.
4. For the **Configuration** setting, select the **Custom** check box to enable the Configuration options.
5. Configure the traffic classifiers as needed.
6. Above the Actions area, select the **Custom** check box to enable the Actions options.
7. For the **Send To** setting, specify what you want the system to do with the traffic related to this application security class. See the online help for assistance with specific screen elements.
8. Click **Finished**.
   The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

Rewriting a URI

You can use the Rewrite URI action to rewrite a URI without sending an HTTP redirect to the requesting client. For example, an ISP provider may host a site that is composed of different web applications, that is, a secure store application and a general information application. To the client, these two applications are the same site, but on the server side they are different applications. Using the Rewrite URI action transparently redirects the client to the appropriate application.

You use Tcl expressions for this setting. If you use a static URI, the system maps the static URI for every incoming request. For details on using Tcl expressions, and Tcl syntax, see the F5 Networks Dev Central web site, **http://devcentral.f5.com**.

Note

*The **Rewrite URI** setting is available only when you select **None** or **Pool** for the **Send To** setting, and you are using the **Hosts** or **URI Paths** traffic classifiers.*

To rewrite a URI

1. In the navigation pane, expand **Application Security** and click **Classes**.
   The HTTP Class list screen opens.
2. Click the **Create** button.
   The New HTTP Class Profile screen opens.
3. Type a name for the application security class.
4. For the **Configuration** setting, select the **Custom** check box to enable the Configuration options.
5. Configure the traffic classifiers as needed, specifically the **Hosts** or **URI Paths** classifiers.
6. Above the Actions area, select the **Custom** check box to enable Actions options.
7. For the **Send To** setting, select **Pool** from the list.
   The screen refreshes and shows more options.
8. For the **Pool** setting, select the name of the local traffic pool to which you want the system to send the traffic.
9. For the **Rewrite URI** setting, type the Tcl expression that represents the URI that the system inserts in the request to replace the existing URI.
10. Click **Finished**.
    The system adds the new application security class, creates a corresponding web application ready for you to configure a security policy, and displays the HTTP Class list screen.

4

Working with Web Applications

- What is a web application?
- Configuring the properties of a web application
- Working with web application groups
- Working with a disabled web application

What is a web application?

In Application Security Manager™, a ***web application*** is the logical representation of the application traffic, defined in the application security class, that you are protecting with a security policy. When you create an application security class, the system automatically creates a corresponding web application for that application. For detailed information on application security classes, refer to Chapter 3, *Working with Application Security Classes*.

You can configure one active security policy for each web application in Application Security Manager. If you have a complex application web site to support, it is possible to create different security policies to protect different parts of the application (such as different security for employee data and customer data). You can create multiple application security classes, thus creating multiple application security web applications and multiple security policies in Application Security Manager, to support one application web site for your company.

Viewing the configured web applications

Once you have created any application security classes, you can review the corresponding list of web applications within the application security configuration. The web applications list provides the following summary information:

- The name of the web application
- The active security policy (if configured), or a link saying **Configure Security Policy**
- The enforcement mode of the security policy
- The logging profile for the web application
- The name of the virtual server, or virtual servers where the HTTP class (with the same name as the web application) was assigned
- The name of the partition the web application is in
- Whether the web application (and the corresponding application security class) is enabled or disabled

Figure 4.1 shows an example of a web application list.

Figure 4.1 *Sample list of web applications*

To view the list of web applications

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. From this screen, you can perform several tasks:
   - Click a web application name to view or modify its properties.
   - Click the **Configure Security Policy** link to start the Deployment wizard.
   - Click an active security policy to view or modify its properties.
   - Click a logging profile to view or modify its properties. Note that you can modify only user-defined logging profiles.

Configuring the properties of a web application

In the Application Security Manager, the web application properties specify the general attributes and preferences for the web application itself. The web application properties help refine how the Application Security Manager processes requests for the web application. The web application properties include:

- Web application language
- Active security policy
- Logging profile

To view the web application properties

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Name column, click a web application name.
   The Web Application Properties screen opens, where you can view and modify the web application's properties, or run the Deployment wizard.

For new, unconfigured web applications, when you click the web application name, the Deployment wizard starts. For more information on working with the Deployment wizard, refer to ***BIG-IP® Application Security Manager™: Getting Started Guide***.

Configuring the web application language

Every web application has a language encoding that determines the character set the browsers use to both display the page and send any form data submitted. The Application Security Manager supports multiple language encodings.

You can configure the language for new web applications (or those you want to reconfigure), and you set the language encoding using the Deployment wizard. F5 Networks recommends that you select the default setting, **Auto detect**, when it is available. The Application Security Manager determines the acceptable character set for the application.

◆ Note

Once you set the web application language, you cannot change it unless you reconfigure the web application completely, losing all settings. For information about reconfiguring web applications, see ***Returning a web application to a new, unconfigured state****, on page 4-5.*

Configuring the active security policy

The ***active security policy*** is the security policy that the Application Security Manager uses to validate requests for, and responses from, the web application. Each web application in Application Security Manager can have only one active security policy at a time.

To configure the active security policy

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Name column, click a web application name.
   The Web Application Properties screen opens.
3. In the Web Application Properties area, from the **Active Security Policy** list, select the security policy that you want to be the active security policy for the web application.
   Note that the system automatically enables (checks) the **Apply Policy** setting when you change the **Active Security Policy** setting on this screen.
4. Click the **Update** button.
   The Web Application List screen reopens, and in the **Active Security Policy** list, you see the Active Policy icon next to the new active security policy.

Note

You can also set the active security policy from most screens in the Configuration utility, in addition to setting it from the Web Application Properties screen, as described above. For more information, see ***Setting the active security policy for a web application****, on page 6-12.*

Specifying the logging profile for a web application

The logging profile determines whether the system logs every request for a web application, logs only those requests that violate the active security policy, or does not log any requests. The logging profile also specifies whether the requests data is stored locally, remotely, or in both places.

You can use a system-supplied logging profile, or you can create a user-defined logging profile. The system provides three logging profiles that you can assign to the web applications:

- Log all requests (locally)
- Log illegal request (locally)
- No logging

If none of these profiles meets your needs, refer to *Configuring logging profiles for web application data*, on page 14-5, for information about how to create logging profiles.

Tip

If your web application receives a high volume of requests, you may want to log only those requests that violate the active security policy so that the system resources are not overburdened. Alternately, you can use remote logging.

To set the logging profile for a web application

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Name column, click a web application name.
   The Web Application Properties screen opens.
3. For the **Logging Profile** setting, select one of the following options:
   - **Log all requests**: Select this option if you want the system to log every request for this web application.
   - **Log illegal requests**: Select this option if you want the system to log only requests which trigger a violation according to the currently active security policy.
   - **No logging**: Select this option if you do not want the system to log any requests for this web application.
   - Optionally, select a user-defined logging profile if one is available.
4. Click the **Update** button.
   The system updates the configuration with any changes you may have made.

Returning a web application to a new, unconfigured state

There may be circumstances when you want to remove all security policies, requests, logging, and configuration information from a web application, and set the web application back to a new, unconfigured state. You can do this by using the **Reconfigure** button on the Web Application Properties screen.

Note

*Using the **Reconfigure** button to clear the configuration information for a web application is a permanent action, and cannot be undone. Use this setting with caution.*

To set a web application back to a new state

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Name column, click the name of a web application that has already been configured.
   The Web Application Properties screen opens.
3. Above the Web Application Properties area on the right, click the **Reconfigure** button.
   A confirmation popup screen opens.
4. Click **OK** to complete the reset action.
   The system deletes all data associated with this web application from the configuration. If you want to secure the web application, you must run the Deployment wizard to create a new security policy.

Working with web application groups

A ***web application group*** is a collection of web applications within the Application Security Manager configuration. Web application groups are made up of two or more web applications. A web application can belong to more than one web application group, however, a web application does not have to belong to a web application group.

The Application Security Manager lists web applications that are not members of any web application group in the **ungrouped** area of the Web Application Groups screen. Recall that there is a one-to-one relationship between application security classes and web applications. In many cases, you may have several application security classes (and thus, web applications) configured for one actual web-based application. You can create a web application group, and then use that group to consolidate the requests, events, and log information about the actual web application.

Creating a web application group

When you create a web application group, you are creating an association between the member web applications. Once you have created a web application group, you can view statistics, logging, and security events in the context of the web application group, in addition to the individual web applications themselves.

To create a web application group

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. On the menu bar, click **Web Application Groups**.
   The Web Application Groups screen opens.
3. Click the **Create** button.
   The Group Properties screen opens.
4. In the **Name** box, type a name for the group.
5. For the **Web Applications** setting, from the **Available** list, select the web applications that you want to add to the new web application group, and use the Move (<<) button to add them to the **Members** list.
6. Click **Save** to update the configuration with the new web application group.

Removing a web application group

If you no longer require the web application group, you can easily remove the group from the configuration. Note that this action does not delete the web applications themselves.

To delete a web application group

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. On the menu bar, click **Web Application Groups**.
   The Web Application Groups screen opens.
3. Check the Select box next to the web application group that you want to delete, and then click **Delete**.
   A confirmation popup screen opens.
4. Click **OK**.
   The system deletes the web application group.

Working with a disabled web application

The Application Security Manager automatically disables web applications when you:

- Disable the **Application Security** setting on an application security class
- Delete an application security class entirely

The system disables the web application because a web application must have a corresponding application security class.

Note

*For more information on application security classes, refer to Chapter 3, **Working with Application Security Classes**.*

Viewing disabled web applications

When the system disables a web application, it moves the web application state from enabled to disabled. You can review the web application state on the Web Application List screen.

To view the disabled web applications

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Web Applications area, in the State column, you can see which web applications are enabled and which web applications are disabled.

Re-enabling a web application

You can re-enable a disabled web application either by creating an application security class with the same name as the disabled web application, or by re-enabling the **Application Security** setting for an existing application security class. In both cases, the system automatically re-enables the disabled web application, as long as the application security class has the same name, exactly, as the disabled web application.

5

Building a Security Policy Automatically

- Overview of automatic policy building
- Configuring automatic policy building
- Viewing the automatic policy building status
- Stopping and starting automatic policy building
- Viewing automatic policy building logs

Overview of automatic policy building

Application Security Manager™ automates the process of creating a security policy to protect a web application. The system must be set up in a networking environment, and be capable of handling traffic to the application.

Here are the primary steps involved in automatic policy building:

- **Set up the security policy**
  First, you use the Deployment wizard to perform initial configuration of a security policy for a web application. Either the Production Site or the QA Lab deployment will initiate automated policy building. You select a policy type to determine which elements to include in the security policy and how to set the rule thresholds. The ***BIG-IP® Application Security Manager™: Getting Started Guide*** describes in detail how to use the Deployment wizard.
- **Let the system automatically add entities to the security policy**
  When the Deployment wizard finishes and traffic is flowing to the application, the system starts the Real Traffic Policy Builder™, the automated policy building tool. The Policy Builder examines requests and responses, populates the security policy with legitimate security policy elements (file types, URLs, parameters, and so on), and puts them in staging if traffic (requests and responses) has the same policy elements, from many sources, over a period of time. If many users encounter the same violations, the violations are likely to be false positives, and the Policy Builder disables the relevant violations or attack signatures.
- **Let the system stabilize the security policy**
  The security policy stabilizes after the system analyzes sufficient traffic, from different sessions, over a period of time. Policy elements are moved out of staging and enforced as they meet the rule thresholds for stabilization.
- **Let the system track site changes and update the policy**
  If the web application changes, the Policy Builder makes the necessary adjustments, and puts the new element in staging. Once stability is reached again, the Policy Builder once again takes elements out of staging and stabilizes the security policy.
- **Review the automatic policy building status**
  On the Automatic Policy Building Status screen, you can review the current status of the security policy, see the policy elements that were added, and view details about the elements and violations listed. If you want more control, you can enforce parts of the security policy from the status screen. The system logs all changes that you or the Policy Builder make to the security policy.

You use the Automatic Policy Building Configuration screen to configure and monitor automatic policy building. The features and settings discussed in this chapter relate directly to the different settings in various areas of the screen.

Configuring automatic policy building

Application Security Manager completely configures the automated policy building settings according to the selections you make when using the Deployment wizard. Configuring these settings may not be necessary in your environment. You can review the settings, and change them if needed.

There are two levels of automated policy building settings: basic and advanced. The basic settings are sufficient for most installations, and require less work. The advanced level allows you to view and change all of the configuration settings if you need further control over security policy details.

Configuring basic automatic policy building settings

Figure 5.1 shows an Automatic Policy Building Configuration screen with the basic settings.

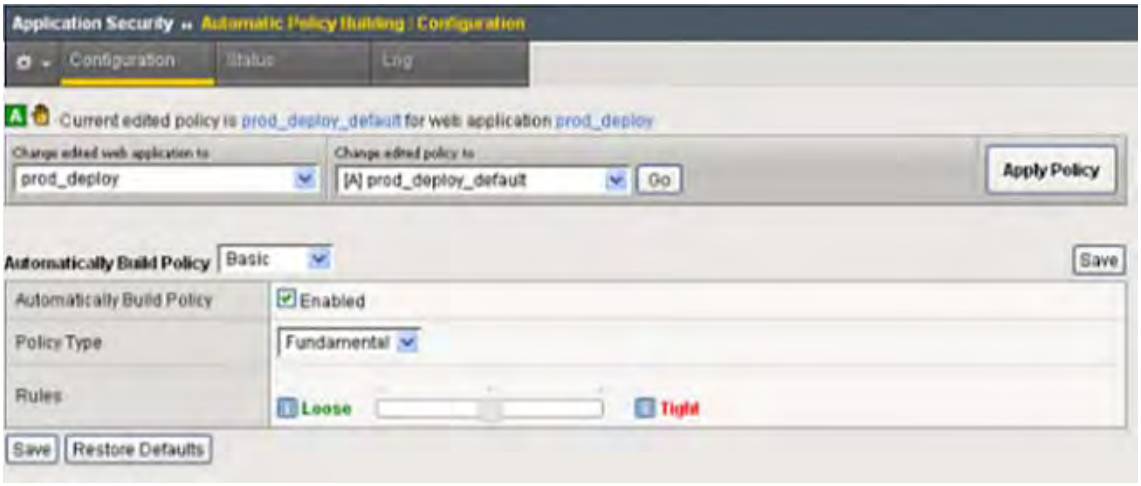


Figure 5.1 *Automatic Policy Building Configuration screen (basic settings)*

To configure basic automatic policy building settings

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For **Automatically Build Policy**, check the **Enabled** box if it is not already checked.
   The screen refreshes and displays more options.

4. For **Policy Type**, select the type of security policy you want to create:
   - **Fundamental**–provides granularity sufficient for most organizations creating a generalized security policy that is easy to maintain. This policy type includes HTTP protocol compliance, evasion techniques, file types and lengths, attack signatures, and the request length exceeds predefined buffer size violation. This is the default setting.
   - **Enhanced**–provides additional granularity and security features suited for customers with higher (and, typically, specific) security needs). This policy type includes all elements in the Fundamental policy type, and also includes parameters and lengths (global level), cookies, and methods.
   - **Complete**–provides the most granular definitions, includes all security features, and is suited for advanced users or customers with extreme security needs. This policy type includes all elements in the Enhanced policy type, and adds URLs and meta characters, parameters (meta characters and URLs), and dynamic parameters (using statistics). This security policy typically takes longer to deploy.
5. For **Rules**, move the slider to change the thresholds of the rules for the security policy:
   - **Loose**
     Builds a security policy using lower threshold values for the rules so they are likely to meet the thresholds more quickly; for example, this setting is useful for smaller web sites with less traffic. Selecting this value may result in more false positives or create a less accurate security policy.
   - **Middle**
     Builds a security policy based on a greater threshold values for the rules. This is the default setting and is recommended for most sites.
   - **Tight**
     Builds a security policy using even higher threshold for the rules and takes longer to meet the thresholds; for example, this setting is useful for large web sites with lots of traffic. Selecting this value may provide fewer false positives and create a more accurate security policy.
6. If you changed any of the settings, click **Save**.
   When traffic is flowing to the application, the system examines requests and responses and begins to build the security policy. This is all you are required to configure unless you want to examine the advanced configuration options. Skip to *Viewing the automatic policy building status*, on page 5-21, for what to do next.

Configuring advanced automatic policy building settings

If you want to review the configuration details of the Policy Builder, you can use the advanced automated policy building settings.

Figure 5.2 shows the Automatic Policy Building Configuration screen with the advanced settings.

Figure 5.2 *Automatic Policy Building Configuration screen (advanced settings)*

To configure advanced policy building settings

If you are already on the Automatic Policy Building Configuration screen, start with step 4.

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For **Automatically Build Policy**, check the **Enabled** box if it is not already checked.
   The screen refreshes and displays more options.
4. To display all configuration options, above the Automatically Build Policy area, select **Advanced**.
   The screen displays the advanced configuration details of the Policy Builder.
5. Review the settings and modify them as needed. Refer to the online help or the following procedures for more information:
   - For details on policy types, see *Changing the policy type*, on page 5-6.
   - For details on security policy elements, see *Modifying security policy elements*, on page 5-9.
   - For details on options, see *Modifying automatic policy building options*, on page 5-11.
   - For details on rules, see *Modifying automatic policy building rules*, on page 5-15.
   - For details on trusted IP addresses, see *Modifying the list of trusted IP addresses*, on page 5-19.
6. If you changed any of the settings, click **Save**.

Changing the policy type

The policy type determines which security policy elements are included in the security policy. When you create a security policy, you can select one of the following policy types:

- **Fundamental** provides security at a level that is appropriate for most organizations, creating a robust security policy, which is highly maintainable and quick to configure. This is the default setting. This policy type includes:
  - HTTP protocol compliance
  - Evasion techniques
  - File types and lengths
  - Attack signatures
  - Request length exceeds predefined buffer size violation
- **Enhanced** provides extra customization, creating a security policy with more granularity. This policy type includes:
  - All options in the Fundamental policy type
  - Parameters and lengths (global level)
  - Cookies
  - Methods
- **Complete** provides an even higher level of customization, creating a security policy with more granularity, but may take longer to configure. This policy type includes:
  - All options in the Enhanced policy type
  - URLs and meta characters
  - Parameters (meta characters and URLs)
  - Parameters at the URL level
  - Dynamic parameters (using statistics)
- **Custom** provides the level of security that you specify when you adjust which security policy elements are included in the security policy. The policy type changes to **Custom** if you change the values from one of the built-in types.

You can change the policy type on the Automatic Policy Building Configuration screen if you want to include a different set of security policy elements in the security policy.

To change the policy type

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For **Automatically Build Policy**, check the **Enabled** box if it is not already checked.
   The screen refreshes and displays more options.
4. For **Policy Type**, select a different type.
   The selected security policy elements and options change depending on the policy type you choose.
5. Click **Save** to save your changes.

Table 5.1 lists each of the security policy elements listed in the Automatic Policy Building configuration, describes what the Policy Builder does when each element is enabled, and shows which policy type enables the element.

Security Policy Element	Description (when enabled)	Policy Type		
		Fundamental	Enhanced	Complete
HTTP Protocol Compliance	Configures the security policy to enable or disable validation checks that ensure HTTP requests are formatted properly.	X	X	X
Evasion Techniques Detected	Configures the security policy to detect evasion techniques and perform normalization processes on URI and parameter input.	X	X	X
File Types	Configures the security policy to add the explicit file types used by the application.	X	X	X
File Types–Lengths	Constructs the security policy to configure length limitations per file type, based on legitimate web application traffic. If you select **Lengths** but not **File Types**, the Policy Builder sets the lengths based on the wildcard (*) file type.	X	X	X
Attack Signatures	Configures the security policy to enable or disable attack signatures.	X	X	X

Table 5.1 *Security policy elements for each policy type*

Security Policy Element	Description (when enabled)	Policy Type		
		Fundamental	Enhanced	Complete
URLs	Configures the security policy to add allowed URLs, based on legitimate traffic.			X
URLs–Meta Characters	Configures the security policy to add allowed meta characters for wildcard URLs, based on legitimate traffic. If you select **Meta Characters** but not **URLs**, the Policy Builder configures the meta characters based on the widlcard (*) URLs for both HTTP and HTTPS.			X
Parameters	Constructs the security policy to add allowed parameters, based on legitimate traffic.		X	X
Parameters–Value Lengths	Constructs the security policy to limit every parameter's length, based on legitimate traffic. If you select **Value Lengths** but not **Parameters**, the Policy Builder adds a parameter (*) wildcard to the security policy and defines its length properties.		X	X
Parameters–Value Meta Characters	Constructs the security policy to add allowed meta characters in parameter values. If you select **Value Meta-Characters** but not **Parameters**, the Policy Builder adds a parameter (*) wildcard to the security policy and defines allowed meta-characters in parameter values.			X
Parameters–Name Meta Characters	Constructs the security policy to add allowed meta characters in parameter names for wildcard parameters. If you select **Name Meta-Characters** but not **Parameters**, the Policy Builder adds a parameter (*) wildcard to the security policy and defines allowed meta-characters in parameter names.			X
Allowed Modified Cookies	Constructs the security policy to add allowed cookies, based on legitimate traffic.		X	X

Table 5.1 *Security policy elements for each policy type (Continued)*

Security Policy Element	Description (when enabled)	Policy Type		
		Fundamental	Enhanced	Complete
Allowed Methods	Constructs the security policy to add allowed methods based on legitimate traffic.		X	X
Request Length Exceeds Predefined Buffer Size violation	Constructs the security policy to enable or disable the **Request length exceeds predefined buffer size** violation.	X	X	X

Table 5.1 *Security policy elements for each policy type (Continued)*

Note that the list in Table 5.1 includes the violations and checks that are relevant only for automatic security policy building. The Application Security Manager includes many other security features that are not included in automatic policy building, such as response scrubbing using Data Guard™, described in Chapter 6, and anomaly detection, described in Chapter 7.

Modifying security policy elements

Security policy elements, such as file types, URLs, evasion technique violations, and so on, form the basis of the security policy that the automatic policy building process is creating. The selected security policy elements are the ones that the Policy Builder configures into the security policy based on legitimate web application traffic.

Each policy type enables a different granularity of policy elements. Refer to Table 5.1, on page 5-7, for a list of policy elements, descriptions of each, and which policy elements are included in each policy type. For example, Figure 5.3 shows the security policy elements that are selected when you select the **Fundamental** policy type on the Automatic Policy Building Configuration screen.

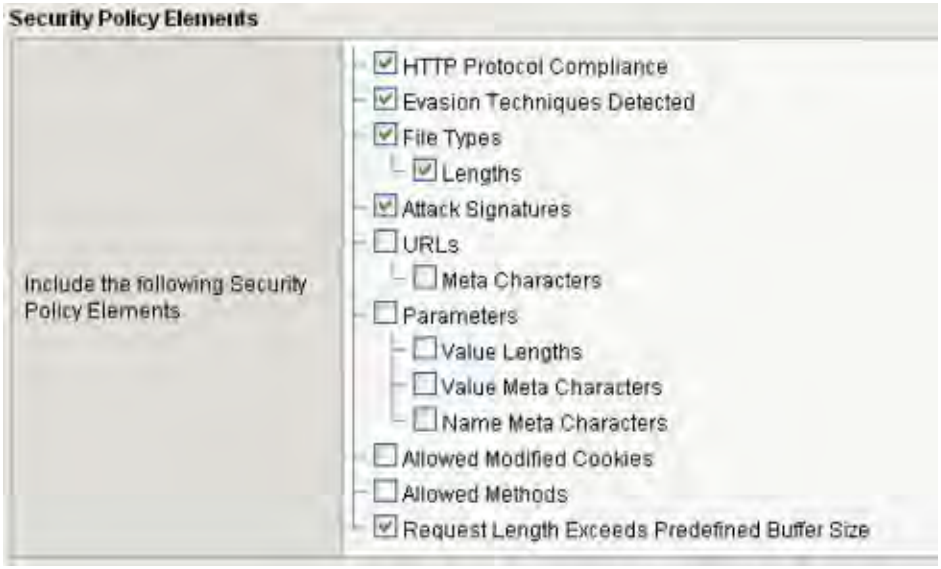


Figure 5.3 *Security policy elements (Fundamental policy type selected)*

You can change the selected policy elements, in which case, the system sets the **Policy Type** to **Custom**.

For file types, URLs, and parameters, if you check the boxes under the element but not the element itself, the system adds a wildcard for the main element and learns the properties you selected.

To modify automatic policy building elements

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For **Automatically Build Policy**, check the **Enabled** box if it is not already checked.
   The screen refreshes and displays more options.
4. To display all configuration options, next from the **Automatically Build Policy** list, select **Advanced**.
5. For **Include the following Security Policy Elements**, select the security policy entities (or violation) that you want the Policy Builder to automatically configure when building the security policy.

   Note: *For file types, URLs, and parameters, you can check the boxes under the element but not the element itself to add a wildcard. For details on the policy elements, see Table 5.1, on page 5-7.*
6. Click **Save** to save your changes.

Modifying automatic policy building options

The Application Security Manager sets the automatic policy building options. These options determine what type of entities the Policy Builder adds to the security policy. You can change the values of the options.

If the web application contains dynamic parameters, you can configure the Policy Builder to identify them. ***Dynamic parameters*** are parameters whose sets of accepted values can change, and usually depend on the user session. For more information on dynamic parameters, refer to *Working with dynamic parameters and extractions*, on page 10-24.

The options also let you simplify your security policy by collapsing similar specific entities into one global entity. After a specified number of occurrences (**10** by default), the system can combine:

- URL-level parameters into a global parameter
- Parameters with similar names into one general name (replacing **param1**, **param2**, and **param3** with **param*)**
- Parameter-specific signature exceptions into a policy-level signature exception

Note

*If you change the values in any of the options, the system sets the **Policy Type** to **Custom**.*

Figure 5.4 shows the Options area of the Automatic Policy Building screen.

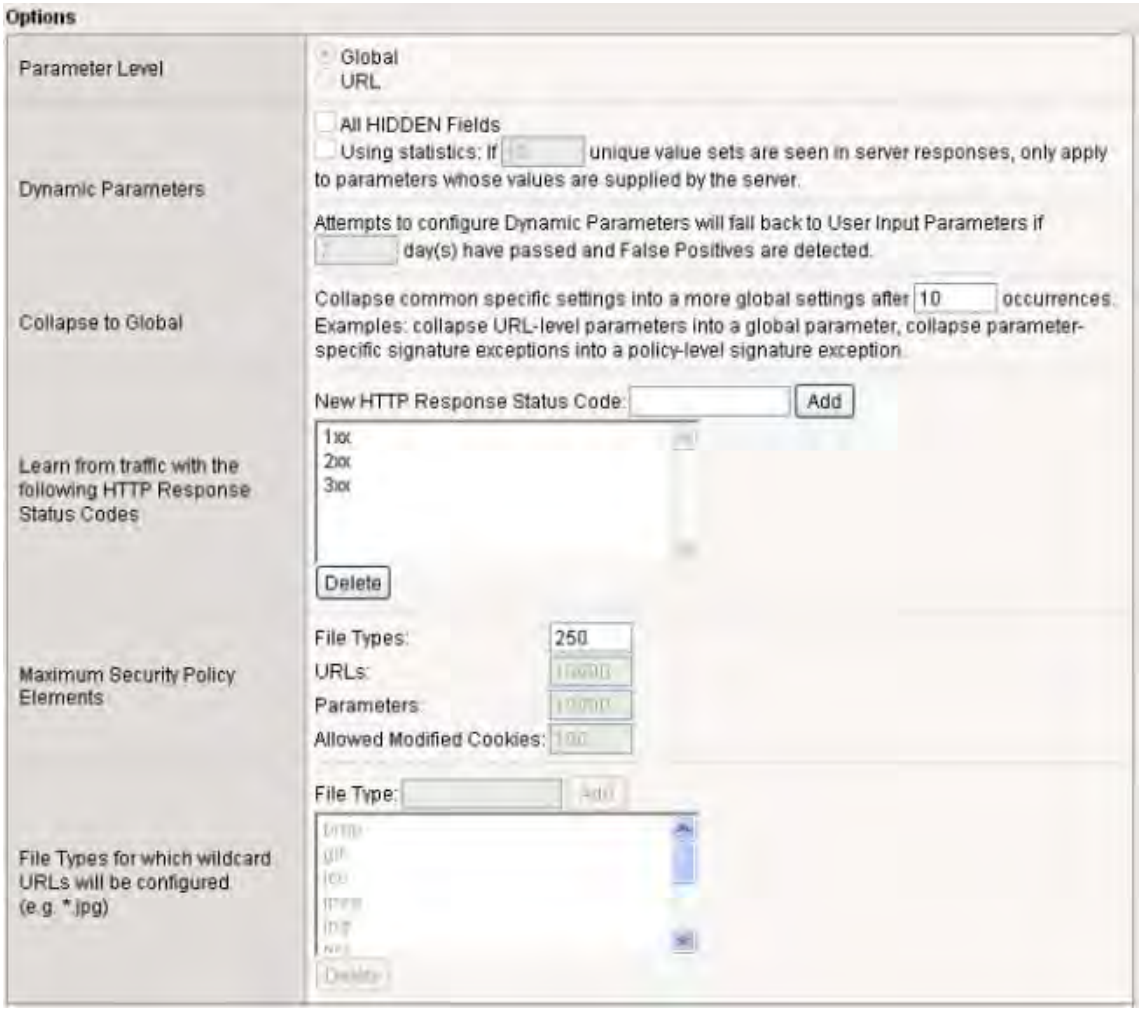

Options
Parameter Level
Global
URL
Dynamic Parameters
All HIDDEN Fields
Using statistics: If unique value sets are seen in server responses, only apply to parameters whose values are supplied by the server.
Attempts to configure Dynamic Parameters will fall back to User Input Parameters if day(s) have passed and False Positives are detected.
Collapse to Global
Collapse common specific settings into a more global settings after 10 occurrences.
Examples: collapse URL-level parameters into a global parameter, collapse parameter-specific signature exceptions into a policy-level signature exception.
Learn from traffic with the following HTTP Response Status Codes
New HTTP Response Status Code:
Add
1xx
2xx
3xx
Delete
Maximum Security Policy Elements
File Types:
250
URLs:
Parameters:
Allowed Modified Cookies:
File Types for which wildcard URLs will be configured (e.g. *.jpg)
File Type:
Add
Delete

Figure 5.4 *Options area on the Automatic Policy Building screen*

To modify automatic policy building options

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. To display all configuration options, next to **Automatically Build Policy**, select **Advanced**.

4. In the Options area, for **Parameter Level**, select how to add parameters to the security policy:
   - To add parameters at the global level, click **Global** (the default value).
   - To add parameters at the URL level, click **URL**.

   Tip: *Both options are available only when both Parameters and URLs are selected in the security policy elements.*
5. Specify whether you want the Policy Builder to add dynamic parameters to the security policy, and if so, where to get them from:
   - If you do not want to include dynamic parameters, make sure both of the dynamic parameters check boxes are cleared, and skip to step 7.
   - To extract dynamic parameters from file types, make sure both the **File Types** and **Parameters** policy elements are already selected in the Security Policy Elements area.
   - To extract dynamic parameters from URLs, make sure the **URLs** and **Parameters** policy elements are selected. Selecting **File Types, Parameters,** and **URLs** also extracts dynamic parameters from URLs.
6. To configure the conditions under which the Policy Builder adds dynamic parameters to the security policy, for **Dynamic Parameters**, perform the following tasks, as needed:
   - To add all hidden form input parameters from the application as dynamic parameters, check the **All Hidden Fields** box.
   - To add dynamic parameters when a number of unique value sets is seen in responses:

     a) Check the **Using statistics** box. This box is checked by default.

     b) Type the number of unique value sets that must be seen for a parameter for the system to consider it a dynamic content value. The default value is **10**.
   - To specify the number of days the parameter must remain unstable before it changes into a user-input parameter, type a number in the box. The default value is **7** days.

     Note*: This number must be longer than the number of days specified in the* ***Stabilize (Tighten)*** *rule, or dynamic parameters will not have enough time to stabilize. For details, see* ***Modifying automatic policy building rules****, on page 5-15.*
7. To simplify your security policy by combining common specific settings into a more global setting, for **Collapse to Global**, type the number of occurrences after which settings are combined.
8. For **Learn from traffic with the following HTTP Response Status Codes**, type the response codes you want to add (for example, add specific codes like **304** or a class of codes like **3xx**). The Policy Builder extracts information from traffic based on transactions that return only those HTTP response status codes.

Tip: *Normally, the Policy Builder learns only from legitimate traffic, so you should add response codes that are returned under normal usage conditions for your application.*

This table shows the correct formats you can use.

Response code	Description
1xx	All informational responses (the request was received; continuing to process it).
2xx	All successful responses (the request was received, understood, accepted, and processed successfully).
3xx	All redirection (the client needs to take additional action on the request).
4xx	All client error responses.
5xx	All server error responses (the server failed to fulfill a request).
Specific codes such as 100, 306, 400, 404	Refer to Hypertext Transfer Protocol -- HTTP/1.1 specification (RFC-2616).

9. For **Maximum Security Policy Elements**, if needed, adjust the maximum number of elements that can be added to the security policy:
   - **File Types** (the default value is **250**)
   - **URLs** (the default is value **10000**)
   - **Parameters** (the default value is **10000**)
   - **Allowed Modified Cookies** (the default value is **100**)

   If the Policy Builder reaches the limit, it stops adding that type of security policy element. If this happens, you may need to intervene:
   - If the web site requires more than the maximum number of elements, you must increase the limits.
   - If the site includes a dynamic element that the Policy Builder cannot learn (such as dynamic sessions in URL or dynamically generated parameter names), either configure the security policy to include the element (for example, dynamic sessions in URL), or clear the element type. The Policy Builder should not be configured to learn that element type in such an environment.
10. For **File Types for which wildcard URLs will be configured**, add the file types for which the Policy Builder adds a wildcard URL instead of adding an explicit URL. Common file types are included by default.

    Tip*: This setting is usually used for static content, such as images, for which a granular policy including every URL is not needed. For example, the Policy Builder adds the wildcard* ****.[Jj][Pp][Gg]*** *instead of* ***image1.jpg****,* ***image2.jpg****, and* ***image3.jpg****.*
11. Click **Save** to save your changes.

Modifying automatic policy building rules

During automatic policy building, the Policy Builder builds security policies in three stages. These stages correspond to the three settings in the Rules area of the Automatic Policy Building Configuration screen. Rules in each stage determine when an element in the security policy moves from one stage to the next. The rules have different values depending on whether the traffic comes from a trusted or untrusted source.

- **Accept as Legitimate (Loosen)**
  During this stage, the Policy Builder identifies legitimate application usage based on seeing repeated behavior from sufficient traffic, over a period of time. The system updates the security policy accordingly. Based on wildcard matches, Policy Builder adds the legitimate policy entities (putting most into staging to learn their properties), and disables violations that are probably false positives.

  For example, when the Policy Builder sees the same file type, URL, parameter, or cookie from enough different user sessions over time, then it adds the entity to the security policy.

- **Stabilize (Tighten)**
  During this stage, the Policy Builder tightens the security policy elements when the rate of security policy changes stabilizes. For example, the Policy Builder enforces an entity type after it records a sufficient number of unique requests and sessions, over a sufficient length of time since the last time an explicit file type, URL, or parameter was added to the security policy.

  Similarly, the Policy Builder enforces the entity's attributes (takes them out of staging) after it records a sufficient number of unique requests and sessions, over a sufficient length of time for a particular file type, URL, or parameter since the last time the entity's attributes or settings were updated.

  When the traffic to the application no longer includes new elements that need to be added to the security policy and the Policy Builder has enforced the policy elements, the security policy is considered stable and its progress reaches 100%.

- **Track Site Changes**
  If a request causes a violation, the Policy Builder looks for changes to the web site. If the Policy Builder discovers changes, it logs the change (Site change detected) and temporarily loosens the security policy to make the necessary adjustments. When the Policy Builder stabilizes the added elements, it retightens the security policy.

  Although it is not recommended, you can disable the **Track Site Changes** option. If you do, when the security policy progress reaches 100% stability, the system disables automatic policy building. The security policy is not updated unless you manually change it, or restart automatic policy building by re-enabling the **Track Site Changes** option.

Figure 5.5 shows the Rules area of the Automatic Policy Building Configuration screen.

Rules
Loose
Tight
Purpose
Rule
Accept as Legitimate (Loosen)
Untrusted Traffic:
For a not-yet-configured policy setting, accept application traffic as legitimate once the system detects matching traffic from
20 different user session(s), spread out over a time period of at least
1 hour(s), but not exceeding
7 day(s).
Trusted Traffic:
For a not-yet-configured policy setting, accept application traffic as legitimate once the system detects matching traffic from
1 different user session(s), spread out over a time period of at least
0 hour(s), but not exceeding
7 day(s).
Stabilize (Tighten)
Untrusted Traffic:
Enforce a policy setting once the system has recorded
5000 total requests, and
500 different user session(s) involving that setting (e.g. for the parameter or file type), spread out over a time period of at least
1 day(s).
Trusted Traffic:
Enforce a policy setting once the system has recorded
5000 total requests, and
500 different user session(s) involving that setting (e.g. for the parameter or file type), spread out over a time period of at least
1 day(s).
Track Site Changes
Enable Track Site Changes
Untrusted Traffic :
For an already-configured policy setting, loosen the setting if the system detects violations from
10 different user session(s), spread out over a time period of at least
5 minute(s), but not exceeding
7 day(s).
Trusted Traffic:
For an already-configured policy setting, loosen the setting if the system detects violations from
1 different user session(s), spread out over a time period of at least
0 minute(s), but not exceeding
7 day(s).

Figure 5.5 *Rules area of the Automatic Policy Building Configuration screen*

Advanced users can view and change the conditions under which the Policy Builder modifies the security policy during any of the three stages. Changing the values in any of the rules (to values not matching any of the built-in levels) also changes the Rules slider to say **Custom** (instead of Loose and Tight).

Note

We recommend that only advanced users change the automatic policy building rule settings only for advanced users. F5 advises using the default values in most cases.

To modify automatic policy building rules

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. To display all configuration options, next to **Automatically Build Policy**, select **Advanced**.
4. In the Rules area, move the slider to change the strictness of the rules for the security policy:
   - **Loose**
     Builds a security policy quickly based on fewer requests; for example, useful for smaller web sites with less traffic.
   - **Middle**
     Builds a security policy based on a medium number of requests. This is the default setting.
   - **Tight**
     Builds a security policy based on a large number of requests; for example, useful for large web sites with lots of traffic.
5. For the **Accept as Legitimate (Loosen)** rule, adjust the number of sessions, and the amount of time that must pass for the Policy Builder to accept and learn a security policy change from traffic. In this stage of security policy building, the Policy Builder adds entities, configures attributes (such as lengths and meta characters), places entities in staging mode, and disables violations.
6. For the **Stabilize (Tighten)** rule, adjust the number of requests, the number of sessions, and the amount of time that must pass for the Policy Builder to stabilize the security policy elements. Stabilizing a security policy element may mean tightening it by deleting wildcard entities, removing entities from staging, and enforcing violations that did not occur.

7. For the **Track Site Changes** rule:

   a) The **Enable Track Site Changes** check box is selected by default. This box must remain checked if you want the Policy Builder to quickly loosen the security policy if changes to the web application cause violations.

   b) Adjust the number of sessions, and the amount of time that must pass for the Policy Builder to update the security policy.
   In this stage of security policy building, the Policy Builder adds wildcard entities, places entities in staging mode, and disables violations.

8. Click **Save** to save your changes.

Modifying the list of trusted IP addresses

You can configure a set of trusted IP addresses for clients that the Policy Builder considers safe in the Trusted IP addresses area of the Automatic Policy Building Configuration screen.

The Policy Builder processes traffic from trusted clients differently than traffic from untrusted clients. For clients with trusted IP addresses, the rules are configured so that the Policy Builder requires less traffic (by default, only 1 user session) to update the security policy with entity or other changes. It takes more traffic from untrusted clients to change the security policy (given the default values).

Figure 5.6 shows the default Accept as Legitimate (Loosen) area of the Automatic Policy Building Configuration screen, configured for a fundamental security policy set to medium strictness. You can see that different values apply to trusted and untrusted traffic.

Rules
Loose
Tight
Purpose
Rule
Accept as Legitimate (Loosen)
Untrusted Traffic:
For a not-yet-configured policy setting, accept application traffic as legitimate once the system detects matching traffic from
20
different user session(s), spread out over a time period of at least
1
hour(s), but not exceeding
7
day(s).
Trusted Traffic:
For a not-yet-configured policy setting, accept application traffic as legitimate once the system detects matching traffic from
1
different user session(s), spread out over a time period of at least
0
hour(s), but not exceeding
7
day(s).

Figure 5.6 *Accept as Legitimate policy building rules for trusted and untrusted traffic*

Refer to *Modifying automatic policy building rules*, on page 5-15, to learn more about how the rules affect the security policy.

To modify the list of trusted IP addresses

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. To display all configuration options, next to **Automatically Build Policy**, select **Advanced**.

4. In the Trusted IP Addresses area, for **IP Addresses**, specify which IP addresses to consider safe:
   - To trust all IP addresses (for internal or test environments), select **All**.
   - To add specific IP addresses or networks, select **Address List**, type the IP address and netmask, then click **Add**.
     The IP address or network range is added to the list. Add as many trusted IP addresses as needed.
   - To delete IP addresses or networks, select the IP address in the list, then click **Delete**.
5. Click **Save** to save your changes.

Restoring default values for automatic policy building

If you change the configuration settings and decide that you want to return them to the system default values, you can change the policy type or use the **Restore Defaults** button.

To restore default configuration values

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. To display all configuration options, next to **Automatically Build Policy**, select **Advanced**.
4. For **Policy Type**, select the type of policy for which you want the default values.
   The screen refreshes and displays the default values for the policy type you selected.
5. Click **Save** to save the default configuration.

You can also click the **Restore Defaults** button at the bottom of the Automatic Policy Building Configuration screen. If you do, the system refreshes and displays the default values for the **Fundamental** policy type.

Viewing the automatic policy building status

You can review the current state of the security policy by looking at the Automatic Policy Building Status screen. A progress bar shows approximately how close the security is to becoming stabilized. You can see a summary of the number of file types, URLs, parameters, and cookies that were added to the security policy.

If you want to understand more about what is happening in the security policy, you can use the Status screen to delve into the details of each policy element. You can override the automatic policy building process and change the security policy before sufficient traffic, sessions, or time has passed.

◆ Note

Overriding the automatic policy building process is for advanced users who are familiar with the web application.

To view the automatic policy building status

1. In the navigation pane, expand **Application Security**, point to **Automatic Policy Building**, then click **Status**.
   The Automatic Policy Building Status screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those for which you want to view the status.
3. To view the number of policy elements that are in the current security policy, review the Policy Elements Learned area. Click the number in the Elements column to examine the specific elements for any entity type.
4. In the Details area, click the expand buttons to show details about the security policy elements. You can make changes to the security policy, if you want, as follows:
   - In the details for **HTTP Protocol Compliance**, **Evasion Techniques Detected**, and **Request Length Exceeds Predefined Buffer Size**, click **Enable** to enforce a check or violation immediately, overriding the rules for adding them.
   - In the stability details for **File Types**, **URLs**, **Parameters**, **Allowed Cookies**, and **Methods**, click **Enforce** to enforce the entity by deleting the entity wildcard (*) from the security policy.
   - In the learning details for **File Types**, **URLs**, **Parameters**, **Allowed Cookies**, and **Methods**, click **Accept** to immediately add specific entities to the security policy, even though they have not met the rules to be accepted as legitimate.
   - In the **Staging** details for **File Types**, **URLs**, and **Parameter**s, click **Enforce** to remove a specific entity from staging, and start enforcing its setting or attributes.
   - In the **Signature stability** details for **Attack Signatures**, click **Enforce** to remove all signatures from staging and enforce them.

- In the learning details for **Attack Signatures**, you can see the list of signatures that the system detected, and which may be false positives. Click **Disable** to remove a signature from staging and disable it.

Figure 5.7 shows the Automatic Policy Building Status screen for a security policy that is still adding policy elements, and is about 25% stabilized. The security policy was developed for trusted traffic, and includes 7 file types, 25 URLs, 32 parameters, and 2 cookies.

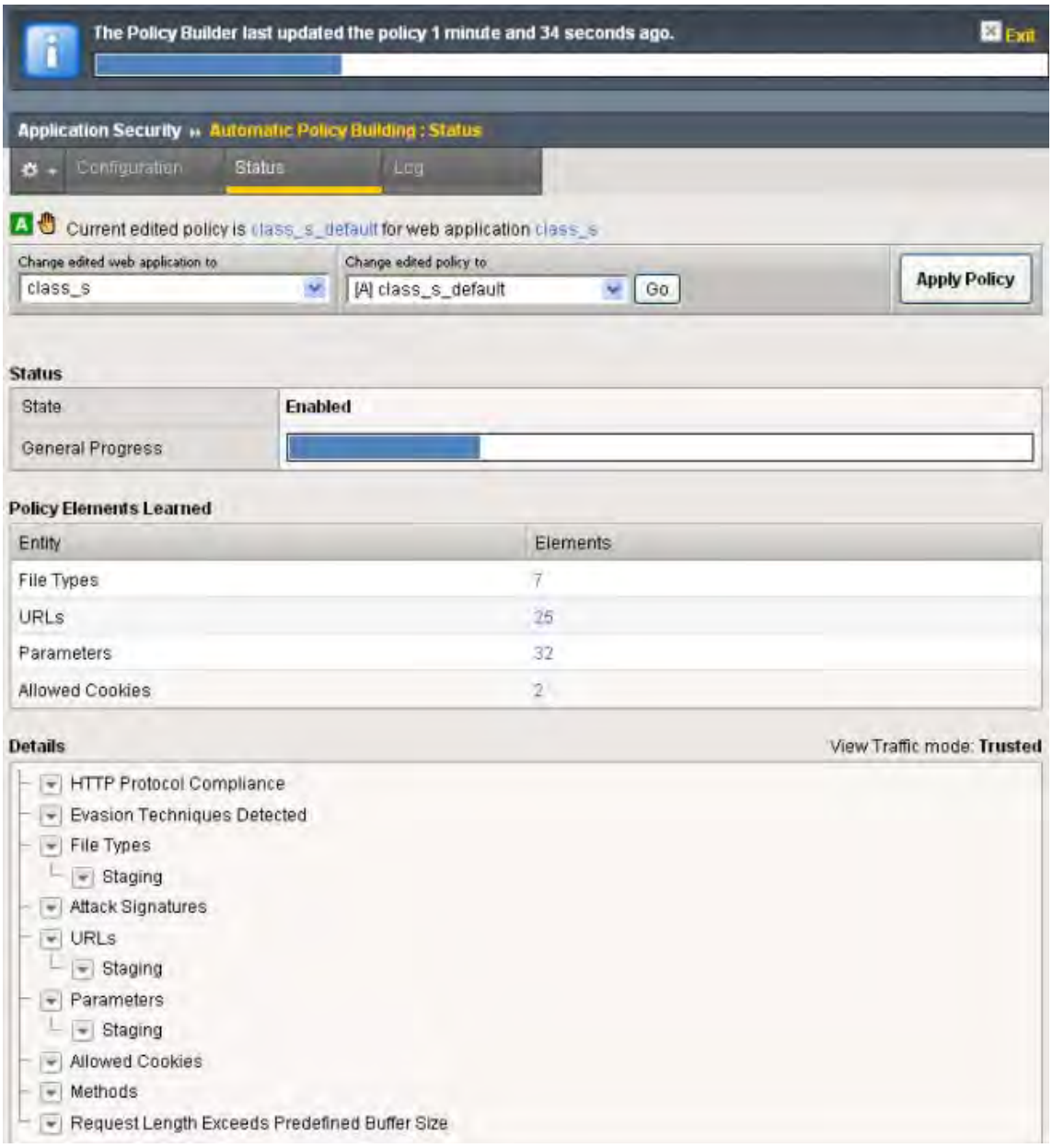


Figure 5.7 *Automatic policy building status*

Stopping and starting automatic policy building

When you use automatic policy building, the Policy Builder can update the security policy as needed, for example, if changes occur on the application web site. You can stop automatic policy building at any time, such as when the security policy stabilizes, and you think the web application will not change for a while.

For security policies that were created using one of the manual methods or imported from an earlier release, you can start automatic policy building, allowing the Policy Builder to add various web site entities to the security policy in order to enhance it.

To stop automatic policy building

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those for which you want to stop automatic policy building.
3. For **Automatically Build Policy**, clear the **Enabled** check box.
   The screen refreshes and shows fewer options.
4. Click **Save** to save the change.
5. On the menu bar, click **Status**.
   The automatic policy building **State** displays **Disabled**, and the system stops the Policy Builder. The security policy remains the same unless you change the configuration manually. Refer to Chapter 6, *Manually Configuring Security Policies*.

To start automatic policy building

1. In the navigation pane, expand **Application Security** and click **Automatic Policy Building**.
   The Automatic Policy Building Configuration screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For **Automatically Build Policy**, check the **Enabled** check box.
   The screen refreshes and shows more options.
4. Click **Save** to save the change.
5. From the menu bar, click **Status**.
   The automatic policy building **State** displays **Enabled**, and the Policy Builder restarts the automatic policy building process based on traffic and configuration settings. Refer to *Configuring automatic policy building*, on page 5-2, if you want to adjust the settings.

Viewing automatic policy building logs

The Application Security Manager creates a log file, called the policy log, for every security policy on the system. This policy log is useful to review changes, or understand when and why the security policy was changed. The automatic policy building policy log includes an entry for each event or action that the Policy Builder makes to the policy.

The system also maintains a second policy log that shows all changes that the Policy Builder or a user made to the security policy. Refer to *Reviewing a log of all security policy changes*, on page 8-9, for how to display it.

To review automatic policy building logs

1. In the navigation pane, expand **Application Security**, point to **Automatic Policy Building**, then click **Log**.
   The Automatic Policy Building Log screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those you are interested in.
3. In the **Filter** area, adjust the filter settings, as needed.
4. Click the **Go** button.
   The screen refreshes, and displays the policy log for the web application and security policy that you selected. Figure 5.8 shows a portion of a sample automatic policy building policy log.
5. In the Description column, click the + magnifying glass to view details about an element that was added to the security policy. For example, see the details for the ***.[Pp][Dd][Ff]** URL in Figure 5.8.

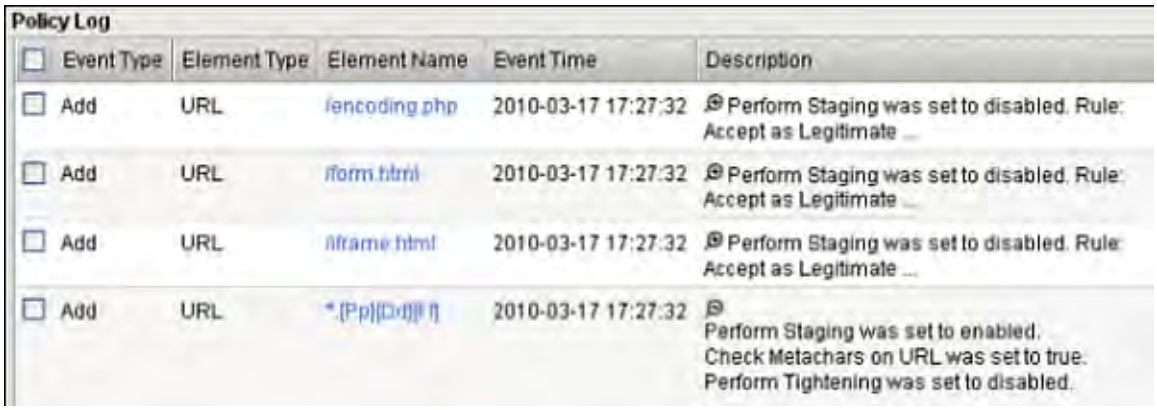


Figure 5.8 *Sample automatic policy building policy log showing changes made by the Policy Builder*

◆ Tip

To display a policy log that shows additional information, such as including manual as well as automatic changes, navigate to the Policy >> Policy Log screen. For details, see ***Reviewing a log of all security policy changes****, on page 8-9.*

6

Manually Configuring Security Policies

- Understanding security policies
- Configuring security policy properties
- Setting the active security policy for a web application
- Validating HTTP protocol compliance
- Adding file types
- Configuring URLs
- Configuring flows
- Masking sensitive data
- Configuring allowed modified cookies
- Configuring mandatory headers
- Configuring allowed methods
- Configuring security policy blocking
- Configuring CSRF protection

Understanding security policies

The core of the Application Security Manager™ security functionality is the security policy, a collection of settings that secures traffic for a web application. The ***security policy*** defines which traffic, including which file types, URLs, and parameters, can access the web application.

When the Application Security Manager receives a request for the web application, the system compares the request to the active security policy. If the request complies with the security policy, the system forwards the request to the web application.

If the request does not comply with the security policy, the system generates a violation (or violations), and then either forwards the request or blocks the request, depending on the enforcement mode of the security policy.

The system similarly checks responses from the web application. Responses that comply with the security policy are sent to the client, but those that do not comply cause violations and may also be blocked.

Creating security policies

You can create security policies using the Deployment wizard. The Deployment wizard builds security policies based on one of several typical deployment scenarios. For information on how to start the Deployment wizard, see *Running the Deployment wizard*, on page 2-5. For details on using the available deployment scenarios, refer to the ***BIG-IP® Application Security Manager™: Getting Started Guide***.

The remainder of this chapter describes the individual configuration tasks that you can perform on screens that relate to existing security policies. If you are using automatic policy building, the Policy Builder performs most of these tasks for you. In that case, refer to Chapter 5, *Building a Security Policy Automatically*.

Configuring security policy properties

The ***policy properties*** are the options and settings that generally define a security policy. You can view and modify the properties of a security policy that you created with the Deployment wizard.

◆ Note

Whenever you change a security policy, you must apply the security policy to make it the active security policy. To remind you that you need to set the active security policy, the system displays an ***[M]*** *next to the modified security policy. After you set the active security policy, the Security Enforcer enforces any changes you made. To set the active policy, refer to* ***Setting the active security policy for a web application****, on page 6-12.*

Configuring the security policy name and description

Each security policy that you configure has a unique name, which you assign as part of the general properties. At minimum, a new security policy must have a name. You can change the security policy name at any time. You can also provide a description of the security policy, to help you better identify the security policy.

To configure the security policy name

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Security Policy Name** setting, type a unique name to replace the existing name.
4. Optionally, in the **Security Policy Description** box, type a description.
5. Click the **Save** button to save any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Viewing the web application associated with the security policy

From the Policy Properties screen, you can easily review the web application associated with a security policy.

To view the web application for a security policy

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited security policy is the one for which you want to view the associated web application properties.
3. In the Configuration area, for the **Web Application** setting, click the web application name link.
   The Web Application Properties screen opens, where you can view the details of the web application.

Configuring the enforcement mode

Security policies can be in one of two enforcement modes:

- **Transparent mode**
  In ***transparent mode***, blocking is disabled for the security policy, and you cannot set the violations to block on the Blocking screen. Traffic is not blocked even if a violation is triggered.
- **Blocking mode**
  In ***blocking mode***, blocking is enabled for the security policy, and you can enable or disable the Block flag for individual violations. Traffic is blocked when a violation occurs, and the system is configured to block that type of violation.

You can set the enforcement mode for a security policy on the Policy Properties screen or the Policy Blocking Settings screen.

When the system receives an incoming request that complies with the security policy, the traffic is always forwarded to the destination, regardless of the mode the security policy is in.

When the system receives an incoming request that does not comply with the security policy, the system generates violations. What happens to the traffic depends on whether the Block flag is set for the violation that occurred. Table 6.1 describes what happens in each mode when an incoming request does not comply with the security policy, and generates a violation.

Enforcement Mode	Block Flag for the Violation That Occurred	Description
Transparent	Enabled	Traffic is sent to the web application.
Transparent	Not enabled	Traffic is sent to the web application.
Blocking	Enabled	Traffic is blocked. The system sends the blocking response page to the client, advises the client that the request was blocked, and provides a support ID number for the violating request.
Blocking	Not enabled (and no other violation with Block enabled occurred)	Traffic is sent to the web application.

Table 6.1 *What happens to the traffic when a violation occurs*

For information on setting the Block flags, refer to *Configuring the blocking actions*, on page 6-43.

To configure the enforcement mode

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Configuration area, for the **Enforcement Mode** setting, select either **Transparent** or **Blocking**.
4. Click **Save** to save any changes you may have made to the security policy properties.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button.

Configuring the staging-tightening period

For each security policy, you can configure the number of days used as the staging-tightening period. Web application entities and attack signatures remain in staging for this period of time before the system suggests that you enforce them. The security policy provides staging suggestions when requests match the attack signatures, or do not adhere to the web application entity's settings.

Note

If the Policy Builder meets the required traffic threshold and runs after the staging-tightening period is over, the Policy Builder automatically enables the web application entities and the attack signatures that did not cause violations during the period.

The system does not enforce wildcard entities when they are in tightening mode. Wildcard entities remain in tightening for the number of days specified by staging-tightening period after which the system suggests you enforce them. In tightening mode, the system adds explicit entities it finds that match these wildcard expressions.

For example, if you enable tightening on file types, the system learns the explicit file types that the web application uses (such as **.html**, **.php**, **.asp**, **.gif**, and .**jpeg**). You can review the new entities and decide which are legitimate entities for the web application, and accept them into the security policy. For more information about the staging-tightening period, see *Understanding staging and tightening for wildcard entities*, on page 9-2.

To configure the staging-tightening period

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Configuration area, for the **Staging-Tightening Period**, type the number of days you want the entities or signatures to be in staging or with tightening enabled. The default value is **7** days.
4. Click **Save** to save any changes you may have made to the security policy properties.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area near the top of the screen.

Enabling or disabling staging for attack signatures

For each security policy, you can enable or disable staging for attack signatures on the Policy Properties screen. The default setting is enabled.

When the staging feature is enabled, the system places all newly assigned and newly updated signatures in staging for the number of days specified in the staging-tightening period. The system does not enforce signatures that are in staging, even if it detects a violation. Instead, the system records the request information. If staging is disabled, the system enforces the signature Learn, Alarm, and Block settings immediately.

For details on configuring attack signatures, refer to Chapter 11, *Working with Attack Signatures*.

To enable or disable signature staging

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Configuration area, specify your staging preference:
   - Check the **Enable Signature Staging** box to enable staging on new or changed signatures. (This is the default setting.)
   - Clear the check box to disable signature staging.
4. Click **Save** to save any changes you may have made to the security policy properties.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area near the top of the screen.

Configuring the maximum HTTP header length

You specify a maximum HTTP header length so that the system knows the acceptable maximum length for the HTTP header in an incoming request. The system applies the length check to header names and value. This setting is useful primarily in preventing buffer overflow attacks.

To configure the maximum HTTP header length

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.

4. For the **Maximum HTTP Header Length** setting, select one of the options:
   - **Any** specifies that the system accepts HTTP headers of any length.
   - **Length** with a value (in bytes) specifies that the system accepts HTTP headers up to that length. The default maximum length is **8192** bytes.
5. Click **Save** to save any changes you may have made to the security policy properties.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring the maximum cookie header length

You specify a maximum cookie header length so that the system knows the acceptable maximum length for any cookie headers in the incoming HTTP request. As with the maximum HTTP header length setting, you can use this setting to help prevent primary buffer overflow attacks.

To configure the maximum cookie header length

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.
4. For the **Maximum Cookie Header Length** setting, select one of the options:
   - **Any** specifies that the system accepts cookie headers of any length.
   - **Length** with a value (in bytes) specifies that the system accepts cookie headers up to that length. The default maximum length is **8192** bytes.
5. Click **Save** to save any changes you may have made to the security policy properties.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring the allowed response status codes

By default, the Application Security Manager accepts all response codes from **1xx** to **3xx** as valid responses. Response codes from **4xx** to **5xx** are considered invalid unless added to the **Allowed Response Status Codes** list. By default, **400**, **401**, **404**, **407**, **417**, and **503** are on the list as valid HTTP response status codes.

If a response contains a response status code from **4xx** to **5xx** that is not on the list, the system issues the violation, **Illegal HTTP status in response**. If you configured the security policy to block this violation, the system blocks the response.

◆ Note

*The Application Security Manager checks only response codes from **400** to **599**. It automatically allows all other response codes.*

To modify allowed response status codes

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.
4. For the **Allowed Response Status Codes** setting, add the response status codes from **4xx** to **5xx** that you want the system to consider legal.
5. Click **Save**.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring dynamic session IDs in URLs

If an application uses dynamic session IDs in URLs, the Application Security Manager cannot use its normal functions to extract and enforce URLs or flows because the URI contains a dynamic element. If the web application that you are securing stores dynamic session ID information in a URL, you can enable the **Dynamic Session ID in URL** option. (In most cases, you do not need to configure this setting.) When the system receives a request in which the dynamic session information does not match the settings in the security policy, the system issues the **Illegal session ID in URL** violation.

When you enable the **Dynamic Session ID in URL** option on the Policy Properties screen (in the Advanced configuration settings), the Application Security Manager extracts the dynamic session information from requests or

responses, based on the pattern that you configure. For requests, the system applies the pattern to the URI up to, but not including, the question mark (**?**) character in a query string.

Using dynamic session IDs does not change the length of the URL with regard to the URL length restriction specified in the file type properties. That is, any length restriction is based on the URL including the session ID.

◆ Note

The system can extract dynamic session information only from URLs that are configured as referrers. See ***Viewing or modifying the properties of a URL****, on page 6-25, for more information.*

To configure dynamic session ID in URL

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.
4. For the **Dynamic Session ID in URL** option, set the option as needed:
   - **Disabled**: The security policy does not enforce dynamic session ID in URL. This is the default value.
   - **Default pattern**: The security policy uses the default regular expression **(\/sap\([^)]+\))** for recognizing a dynamic session ID in URL.
   - **Custom pattern**: The security policy uses a user-defined regular expression to recognize a dynamic session ID in URL. Type a regular expression in the **Value** box, and a description in the **Description** box.
5. Click **Save**.
   The system updates the configuration with any changes you have made.

Activating iRule events

An ***iRule*** is a script that lets you customize how you manage traffic on the BIG-IP system. You can write iRules™ to modify a request or response based on violations that occur. For detailed information on writing iRules, see the F5 Networks DevCentral web site, **http://devcentral.f5.com**.

If you want to use iRules to activate specific Application Security Manager iRule events that iRules can subscribe to and perform actions based on, you must enable the **Trigger ASM iRule event** setting. By default, the iRule event check box is cleared. Table 6.2 lists the iRule events that iRules can subscribe to in Application Security Manager.

Application Security iRule Event	Description
ASM_REQUEST_VIOLATION	Occurs when Application Security Manager detects a request that violates a security policy.
ASM_REQUEST_BLOCKING	Occurs when Application Security Manager is generating an error response to the request that caused the violation, and gives the iRule a chance to modify the response before it is sent.
ASM_RESPONSE_VIOLATION	Occurs when Application Security Manager detects a response that violates a security policy.

Table 6.2 *Application Security Manager iRule events*

To activate iRule events

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.
4. If you have written iRules to process application security iRule events, and assigned them to a specific virtual server, check the **Trigger ASM iRule event** box.
5. Click **Save**.
   The system updates the configuration.

Configuring trusted XFF headers

You can configure Application Security Manager to trust XFF (X-Forwarded-For) headers or customized XFF headers in requests. You may want to configure trusted XFF headers if the Application Security Manager is deployed behind an internal or other trusted proxy. Then, the system uses the IP address that initiated the connection to the proxy instead of the internal proxy's IP address. This option is useful for logging purposes and for the geolocation feature.

You should not configure trusted XFF headers if you think the HTTP header may be spoofed, or crafted, by a malicious client.

To configure trusted XFF headers

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes, and displays additional configuration options.
4. Check the **Trust XFF Header** box.
   The screen refreshes, and displays custom XFF configuration options.
5. If your web application uses custom XFF headers, add them as follows:
   a) For **New Custom XFF Header**, type a XFF header that the system should trust.
   b) Click **Add**.
   You can configure up to five custom XFF headers.
6. Click **Save**.
   The system updates the configuration.

Setting the active security policy for a web application

At any given time, the Application Security Manager enforces only one security policy for a web application. The security policy that is currently protecting the web application is called the ***active security policy***. The active security policy is marked with the Active icon in the Security Policies List for the web application.

To activate a security policy

1. In the navigation pane, expand **Application Security** and click **Web Applications**.
   The Web Application List screen opens.
2. In the Name column, click the name of the web application for which you are activating a security policy.
   The Web Application Properties screen opens.
3. In the **Active Security Policy** list, select the security policy that you want to be the active security policy for the web application. Note that the system automatically checks the **Apply Policy** setting when you change the **Active Security Policy** setting on this screen.
4. Click the **Update** button.
   The system changes the active security policy.
5. To verify the change, In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens, and you see the Active Policy icon next to the new active security policy.

Tip

*You can also change the active security policy from most of the screens throughout the Application Security Manager. Change the edited policy then click **Go** in the editing context area.*

Determining when to set the active security policy

When you change the configuration in a security policy, you must apply the security policy before the changes take effect. The following icons indicate the state of a security policy.

- The Active icon A next to a security policy name indicates the active security policy. You may also see an A in square brackets **[A]** to indicate the active security policy. Only one security policy can be the active security policy.
- The Modified icon M or **[M]** next to a security policy name indicates that the security policy has been modified. Clicking **Apply Policy** enforces the changes and removes the icon.

Figure 6.1 shows a Security Policies list containing two policies. The security policy called **webapp_security** is the active policy and it has been modified.

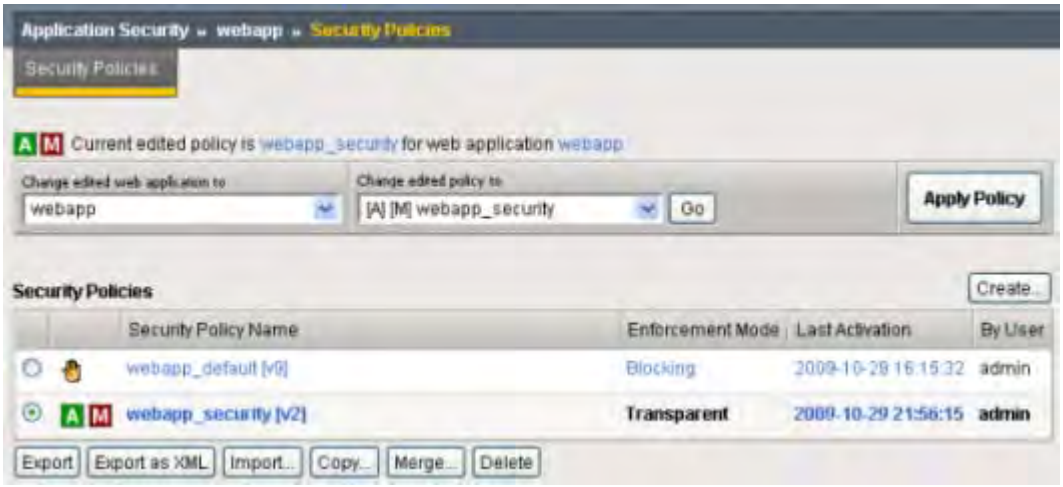


Figure 6.1 *Security Policies list*

Validating HTTP protocol compliance

The first security checks that Application Security Manager performs are those for RFC compliance with the HTTP protocol. The system performs validation checks on HTTP requests to ensure that the requests are formatted properly. For each security policy, you can configure which HTTP protocol checks the system performs, and what happens if requests are not HTTP compliant.

Requests that fail any of the enabled protocol checks trigger an **HTTP protocol compliance failed** violation. You can configure the system to generate learning suggestions, alarms, or block requests that cause the violation. The system blocks requests that are not compliant with HTTP protocol standards if the security policy enforcement mode is set to blocking, and the system is configured to block traffic if the violation occurs.

Understanding how HTTP protocol validation affects application security checks

When Application Security Manager receives a request from a client, the first aspect of the request that the system validates is HTTP protocol compliance. In rare cases, if a request triggers one of the following HTTP protocol subviolations and blocking is disabled for that subviolation, Application Security Manager may stop evaluating the request further and send it to the server. If you keep these subviolations in blocking mode, which they are by default, this situation should never happen. The HTTP protocol validations that may cause this situation are:

- Unparsable request content
- Null in request–except Null in binary part of multipart, chunked request
- Several content-length headers
- Bad multipart/form-data request parsing (not checked by default)
- Bad multipart parameters parsing
- Bad HTTP version (not 1.0 or 1.1)

In most cases, requests not meeting these validations contain payloads that most likely will not be parsed by the application, nor clearly indicate a malicious action.

Note

*If a request is too long and causes the **Request length exceeds defined buffer size** violation, the system stops validating that request.*

Configuring HTTP protocol compliance validation

You can review and modify the validation options for HTTP protocol compliance.

If you use automatic policy building, the system immediately enables the **Learn**, **Alarm**, and **Block** settings for the **HTTP protocol compliance failed** violation but does not enable any of the HTTP protocol checks. After the system has processed sufficient traffic from different users over a period of time, it enables the appropriate HTTP protocol checks.

To view the HTTP protocol validation options

1. In the navigation pane, expand **Application Security**, point to **Policy**, then click **Blocking**.
   The Security Policy Blocking Settings screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the RFC Violations area, for the **HTTP protocol compliance failed** violation, review and, if needed, modify the **Learn**, **Alarm**, and **Block** settings.
4. Click the **HTTP protocol compliance failed** violation.
   The HTTP Protocol Compliance screen opens.
5. On the HTTP Protocol Compliance screen, enable or disable the HTTP protocol checks, as required. For an explanation of the individual validations, refer to the online help.
6. Click **Save** to retain any changes you made.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Adding file types

Using the Allowed File Types screen, you can specify the file types that are allowed (or disallowed) in the web application:

- **Wildcard file type**
  Wildcard file types are those whose name is, or contains, a pattern string. When you configure a wildcard file type, and enable tightening, as the security policy processes traffic, the system discovers the file types that match the wildcard. You can then decide whether to add those file types to the security policy. For detailed information on wildcard file types, refer to *Configuring wildcard file types*, on page 9-5.
- **No extension file type**
  The no extension file type represents file types that do not have the typical file extension as part of the name. The slash character (/) is an example of a **no_ext** file type.
- **Explicit file type**
  Explicit file types have a known file extension name, for example, **JSP** or **htm**.
- **Disallowed file types**
  You can also configure a list of file types that the system always rejects. These objects are known as ***disallowed file types***. Refer to *Disallowing specific file types*, on page 6-20, for more information.

Note

*File types are case-sensitive. As a result, the security policy processes **JPG** and **jpg** files as separate file types.*

You can build the list of allowed file types in the security policy in these ways:

- You can run the Policy Builder. See Chapter 5, *Building a Security Policy Automatically*, for more information.
- You can enforce an allowed file type from the Allowed File Types list. See *Adding new entities to the security policy from staging or tightening*, on page 13-13.
- You can accept an allowed file type from a learning suggestion. See *Accepting a learning suggestion*, on page 13-7.
- You can manually add each file type, as explained in this section.

Note

*When using automatic policy building, the system automatically creates a **no_ext** file type for URLs with no file extension and URLs with file extensions longer than eight characters.*

Creating allowed file types

For ***allowed file types***, which are file types that the system accepts, you can configure lengths, and whether to check responses for the associated requests. Table 6.3 describes the allowed file type properties.

Allowed file type property	Description
File Type	Specifies a file type definition that allows the file types it defines. The file type definition can be for either a unique explicit file type or a wildcard definition. File types are case-sensitive. The available file types are: **Explicit:** Specifies a unique file type name. Type the file type name in the adjacent box. **Wildcard:** Specifies that the file type is a wildcard expression. Any file type that matches the wildcard expression is considered legal. For example, entering the wildcard [*] specifies that the security policy allows any file type. Type a wildcard expression in the adjacent box. **No Extension:** Specifies that the web application has a URL with no file type. The system automatically assigns this file type the name **no_ext**.
Perform Staging	Specifies, when checked, that the system places this entity in staging. Staging can be applied to both explicit and wildcard file types. If an entity is in staging, the system does not block requests for this entity even when a violation (such as file type length) occurs and the security policy is in blocking mode. The system logs learning suggestions produced by the requesting staged entities on the Learning screens. You can check the staging status on the Allowed File Types screen. If a file type is in staging, the system displays a light bulb icon (in different colors indicating status). Move the cursor over the light bulb icon to display staging information. When the file type has been in staging for the staging period and you are no longer getting learning suggestions, you can clear this check box. ***Note**: F5 Networks recommends against using both tightening and staging on the same wildcard entity.*
Perform Tightening	Specifies, when checked, that tightening is enabled for this wildcard file type. Tightening is only relevant for wildcard entities. As a result, -When Policy Builder runs, it adds explicit file types that do not exist in the security policy but match this wildcard. -The Staging-Tightening Summary screen shows how many entities are in staging or with tightening enabled. You can review the explicit file types that do not exist in the security policy but match this wildcard file type, decide which are legitimate for the web application, and accept them into the security policy. ***Note**: F5 Networks recommends against using both tightening and staging on the same wildcard file type.*
URL Length	Specifies the acceptable length, in bytes, for a URL in the context of an HTTP request containing this file type.
Request Length	Specifies the maximum acceptable length, in bytes, for the whole HTTP request that applies to this file type.
Query String Length	Specifies the maximum acceptable length, in bytes, for the query string portion of a URL that contains the file type.

***Table 6.3** File type properties*

Allowed file type property	Description
POST Data Length	Specifies the maximum acceptable length, in bytes, for the POST data of an HTTP request that contains the file type.
Check Response	Specifies that the system enables response filtering by attack signatures that are designed to inspect server responses.

Table 6.3 *File type properties (Continued)*

To manually create an allowed file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed File Types list, click the **Create** button.
   The Add Allowed File Type screen opens.
4. For the **File Type** setting, select the type, and then type a file extension or wildcard expression.

   If you select **No Extension**, the system specifies **no_ext**.

   Tip*: For more information about wildcard file types, see* ***Configuring wildcard file types****, on page 9-5.*
5. For the length settings, specify the appropriate values. This is optional.
6. If you want the system to validate responses for this file type, check the **Check Response** box.
7. Click the **Create** button.
   The screen refreshes, and you see the new file type on the Allowed File Types screen.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Modifying file types

You can modify any of the file type flags, or characteristics, depending on the needs of the web application.

To modify the allowed file type characteristics

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed File Types list, click the name of the file type that you want to update.
   The Edit Allowed File Type screen opens.
4. Make any changes as required, and click the **Update** button.
   The screen refreshes, and returns to the Allowed File Types screen.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Removing file types

Since web applications can change on a regular basis, you may find that the file types list contains file types that an application should not have. You can remove the file type and any related URLs (if applicable) in one step.

To remove an allowed file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed File Types list, click the Select box to the left of the file type that you want to remove from the list.
4. Click the **Delete** button below the list.
   The system removes the file type from the configuration.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Disallowing specific file types

For some web applications, you may want to deny requests for certain file types. In this case, you can create a set of disallowed file types. When the Application Security Manager receives a request whose file type is disallowed, the system ignores, learns, logs, or blocks these illegal file types according to the settings you configure for the **Illegal File Type** violation on the Blocking Policy screen. Adding disallowed file types is useful for file types you know should never appear on your site (such as .**exe** files) or for files on your site that you never want users from the outside to reach (such as .**bak** files).

To disallow a file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. On the menu bar, click **Disallowed File Types**.
   The Disallowed File Types screen opens.
3. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
4. Above the DisAllowed File Types list, click the **Create** button.
   The New Disallowed File Types screen opens.
5. For the **File Type** setting, type the file type that you want to disallow (for example, **jpg** or **exe**).

   ***Note**: File types are case-sensitive.*
6. Click the **Create** button.
   The screen refreshes, and displays the updated Disallowed File Types screen.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring URLs

You can add three types of ***URLs*** for the web application that you are protecting:

- **Explicit URLs**
  An ***explicit URL*** has a specific name and represents one file or component of the web application, for example, **/login.jsp** or **/sell.php**.
- **Wildcard URLs**
  A ***wildcard URL*** is one whose name is or contains a pattern string, for example, ***** or ***.png**. For more information on managing wildcard URLs, refer to *Configuring wildcard URLs*, on page 9-9.
- **Disallowed URLs**
  A disallowed URL is a URL that is not allowed by the security policy. For information on creating disallowed URLs, refer to *Configuring URLs not allowed by the security policy*, on page 6-26.

Table 6.4 lists URL properties.

URL property	Description	Applies to
URL	Specifies a URL definition that allows the URLs it defines. The URL definition can be for either a unique explicit file type or a wildcard definition. URLs are case-sensitive. The available types are: **Explicit:** Specifies that the URL is a unique URL. Type the URL in the adjacent box. **Wildcard:** Specifies a wildcard expression. Any URL that matches is considered legal. For example, typing * specifies that any URL is allowed by the security policy. Type a wildcard expression in the adjacent box.	Explicit URLs and wildcard URLs
Perform Staging	Specifies, when checked, that the system places this URL in staging. When in staging, the system does not block **Illegal meta character in URL** violations. Learning suggestions produced by requesting staged URLs are logged in the Learning screens. You can check the staging status on the URL List screen. If a parameter is in staging, the system displays a light bulb icon (in different colors indicating status). Move the cursor over the light bulb icon to display staging information. When the URL has been in staging for the staging period and you are no longer getting learning suggestions, you can clear this check box. ***Note**: F5 Networks recommends against using both tightening and staging on the same wildcard entity.*	Wildcard URLs only

***Table 6.4** URL properties*

URL property	Description	Applies to
Perform Tightening	Specifies, when checked, that tightening is enabled. As a result: -When Policy Builder runs, it adds explicit URLs that do not exist in the security policy but match this wildcard URL. -The system displays, on the Staging-Tightening Summary screen, how many entities are in staging and/or with tightening enabled. You can review the explicit URLs that do not exist in the security policy but match this wildcard URL, decide which are legitimate for the web application, and accept them to the security policy. Specifies, when cleared, that the Policy Builder does not add to the security policy explicit URLs that match this wildcard URL, and the system does not suggest URLs that match this wildcard URL. The default is disabled. ***Note**: F5 Networks recommends against using both tightening and staging on the same wildcard URL.*	Wildcard URLs only
Protocol	Specifies whether the protocol for the URL is HTTP or HTTPS.	Explicit URLs, wildcard URLs, and disallowed URLs
Check Flows to this URL	Specifies, when checked, that the security policy validates the flows to the URL. If this setting is disabled, the Security Enforcer ignores the flows to the URL. For more information on flows, refer to *Configuring flows*, on page 6-30. When you check this box, additional settings appear.	Explicit URLs only
URL is Entry Point	(Visible when **Check Flows to this URL** is selected.) Specifies, when checked, that this URL is a page through which a visitor can enter the web application.	Explicit URLs only
URL is Referrer	(Visible when **Check Flows to this URL** is selected.) Specifies, when checked, that the URL is a URL from which a user can access other URLs in the web application.	Explicit URLs only
URL can change Domain Cookie	Specifies, when checked, that the security policy does not block an HTTP request where the domain cookie was modified on the client side. Note that this setting is applicable only if the URL is a referrer.	Explicit URLs only
Apply XML Profile	Specifies, when checked, that the system validates XML data found in requests to this URL. The default is disabled. For more information on XML security, refer to Chapter 12, *Protecting XML Applications*.	Explicit URLs and wildcard URLs
XML Profile	(Visible when **Check XML** is selected.) Specifies that the system validates XML data found in requests to this URL based on the settings you configure in a specific XML profile. For more information on XML profiles, refer to *Associating an XML profile with a URL*, on page 12-20.	Explicit URLs and wildcard URLs

***Table 6.4** URL properties (Continued)*

URL property	Description	Applies to
Check XML Content-Type Headers	(Visible when **Check XML** is selected.) Specifies the kind of information the XML profile is to protect. • **All** specifies that the system validates XML data found in requests to this URL. • **User defined** specifies that the system validates XML data found in requests to this URL only if the context-type header includes a specific string.	Explicit URLs and wildcard URLs
Check AMF (When the content type matches “amf”)	(Visible only when **Check XML** is selected.) Specifies, when checked, that the system applies security checks to Action Message Format (AMF) requests. For more information, refer to *Configuring AMF security for URLs*, on page 6-27.	Explicit URLs and wildcard URLs
URL Description	Describes the URL (optional).	Explicit URLs and wildcard URLs
Check characters on this URL	Specifies, when enabled, that the system verifies meta characters on this URL.	Wildcard URLs only

Table 6.4 *URL properties (Continued)*

Overview of URL parameters and extractions

URL parameters are parameters that are associated with a specific URL. ***Extractions*** specify how the system discovers dynamic parameters and their properties. For full details on managing URL parameters and extractions, refer to *Working with dynamic parameters and extractions*, on page 10-24.

Overview of URL flows

Flows are the navigational relationships between the entities in a web application. Configuring flows may tighten the security policy, but this is an optional configuration option. For more information on flows, refer to *Configuring flows*, on page 6-30.

Creating an explicit URL

You can build the list of URLs in the security policy in these ways:

- You can run the Policy Builder. See Chapter 5, *Building a Security Policy Automatically*, for more information.
- You can enforce a URL from the Allowed URLs list. See *Adding new entities to the security policy from staging or tightening*, on page 13-13.
- You can accept a URL from a learning suggestion. See *Accepting a learning suggestion*, on page 13-7.
- You can manually add each URL to the security policy, as explained in the following procedure.

To create an explicit URL manually

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed URLs List area, click the **Create** button.
   The New Allowed URL screen opens.
4. To configure the advanced settings, select **Advanced** above the Create New Allowed URL area.
   The screen refreshes to display additional settings.
5. In the Create New Allowed URL area, for the **URL** setting, select the type, and then type the explicit URL name. For explicit URLs, be sure to type the full resource path starting with the slash (/) character.
6. In the **Protocol** list, select the protocol to be used to access the URL.
7. Apply the XML profile and configure XML settings, if needed. For information on checking XML data, refer to *Associating an XML profile with a URL*, on page 12-20. For information on checking AMF data, refer to *Configuring AMF security for URLs*, on page 6-27.
8. Click the **Create** button.
   The screen refreshes, and you can see the new URL in the list.
9. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

To display URLs visually, you can display a tree view of the security policy that shows the explicit URLs with any associated parameters. For more information on the tree view, refer to *Displaying security policies in a tree view*, on page 8-10.

Removing a URL

Web applications can change over time. Therefore, you may want to remove obsolete URLs from the security policy.

To remove a URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, check the box to the left of the URLs you want to remove.
4. Click the **Delete** button.
   A confirmation popup screen opens, where you confirm the deletion of the URL.
5. Click **OK**.
   The system removes the URL from the security policy.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Viewing or modifying the properties of a URL

You can review and modify the properties of an individual URL.

To view or modify a URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, click the name of a URL.
   The Allowed URL Properties screen opens, where you can view or modify the URL properties.

◆ Tip

If the URL name is in gold letters, the URL is a referrer. Referrers call other URLs within the web application. See ***Identifying referrer URLs****, on page 6-26, for more information.*

Identifying referrer URLs

In lists of URLs, non-referrer URLs appear in blue and referrer URLs appear in gold. ***Referrer URLs*** are web pages that can request other URLs. For example, an HTML page can request a GIF, JPG, or PNG image file. The HTML page is the referrer, and the GIF, JPG, and PNG files are non-referrers.

A referrer in Application Security Manager is similar to the HTTP Referer header. If you need to configure referrers, use them for complex objects, such as HTML pages, but not for embedded objects, such as GIF files.

Figure 6.2 illustrates a referrer URL (**/register.php**) and a non-referrer URL (/login.php).

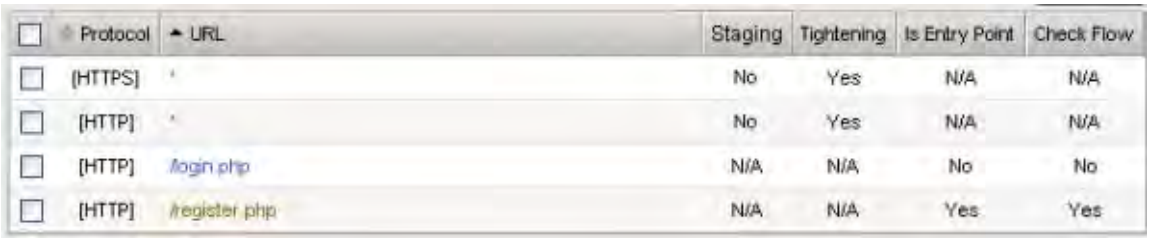


Figure 6.2 *Example of a referrer URL (/register.php)*

Configuring URLs not allowed by the security policy

You can create a list of disallowed URLs, for example, to disallow access to an administrative interface to the web application such as by disallowing **/admin/x.php**. If a requested URL is on the disallowed URLs list, the system ignores, learns, logs, or blocks these illegal URLs according to the settings you configure for the **Illegal URL** violation on the Blocking Policy screen. You can view learning suggestions for disallowed URLs on the Illegal URL learning screen. For more information, refer to *Working with learning suggestions*, on page 13-2.

To add disallowed URLs

1. In the navigation pane, expand **Application Security**, point to **URLs**, and then click **Disallowed URLs List**.
   The Disallowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are the ones you want to update.
3. Click the **Create** button.
   The New Disallowed URL screen opens.
4. For **Protocol**, select **HTTP** or **HTTPS**.
5. For **URL**, type the name of the URL you want the security policy to consider illegal in the format **/index.html**.

6. Click the **Create** button.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring AMF security for URLs

Action Message Format (AMF) is a binary format that is loosely based on Simple Object Access Protocol (SOAP). AMF is used primarily to exchange data between Adobe® Flash® applications and a database, by using the RPC (remote procedure call) protocol. The Application Security Manager secures applications that use AMF by applying attack signatures to parameters within POST requests, when the **Content-Type** header matches the pattern ***amf***.

Determining if a request is an AMF request

The Application Security Manager determines whether a request is an AMF request when the request meets the following criteria:

- The **Content-Type** header matches the pattern ***amf***
- The requested URL (wildcard or explicit) is in the security policy, and the **Check AMF** option is enabled for that URL
- The request has a POST body

When a request matches the previous criteria, the system applies parameter-specific attack signatures to any parameters in the POST body of AMF requests. If the system finds a match, then the Security Enforcer issues the **Attack signature detected** violation, and handles the request according to the blocking policy settings.

◆ Note

*If a request contains a **Content-Type** header whose value matches the ***amf*** pattern, but the **Check AMF** option is not enabled for the corresponding URL, then the Application Security Manager does not apply the additional AMF checks.*

Configuring AMF security checks

You configure AMF security on a per-URL basis. You can configure AMF security for both wildcard URLs and explicit URLs.

Note

The following procedure is for configuring AMF security for a URL that already exists in the configuration. If the URL does not yet exist, refer to ***Creating an explicit URL****, on page 6-24, or* ***Creating wildcard URLs****, on page 9-9, before proceeding.*

To configure AMF security

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, click the name of the URL for which you want to enable AMF security.
   The Allowed URL Properties screen opens.
4. Above the Allowed URL Properties area, select **Advanced**.
   The screen refreshes, and displays an additional option.
5. Check the **Check AMF (When the content type matches "amf")** box.
6. Click the **Update** button.
   The screen displays the Allowed URLs List screen.

Working with the URL character set

When you use the Deployment wizard to create a security policy, you select a language encoding. The system then enforces the character set of that language encoding in the URL element in URIs, and also for any wildcard URLs in the security policy. For example, by disallowing the characters <, >, ', and |, Application Security Manager can protect against many cross-site scripting attacks and injection attacks. You can modify which characters are enforced in the URL character set.

Note

You can also configure which characters are allowed in parameters. See ***Working with the parameter character sets****, on page 10-29, for more information.*

To view or modify the character set for URLs

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. On the menu bar, click **Character Set**.
   The URLs Character Set screen opens, where you can view the character set, and state of each character.
4. Use the **Filter** option to display only those characters that you want to view.
5. To modify the character set, click **Allow** or **Disallow** to define which characters the system should permit or prohibit in the name of a wildcard URL.
6. Click **Save** to save your changes.
7. Click the **Apply Policy** button (in the editing context area) to immediately put those changes into effect.

◆ Tip

*To restore the default character set definitions, you can click the **Restore Defaults** button at any time.*

Configuring flows

The ***application flow*** defines the access path leading from one URL to another URL within the web application. For example, a basic web page may include a graphic and a hyperlink to another page in the application. The calls to these other entities from the basic page make up the flow.

◆ Note

Configuring flows is an optional task. Unless you need the enhanced security of configured flows, F5 Networks recommends that you do not configure flow-based security policies due to their complexity.

Viewing the entire application flow

You can view the application flow in its entirety, or you can view the flow for an individual URL.

To view the entire application flow

1. In the navigation pane, expand **Application Security** and click **Flows**.
   The Flows List screen opens.
2. In the Flows List area, click the arrow to see the flows from this URL or list of entry points flow.

Viewing the flow to a URL

When you view the flows for a particular URL, the system displays the flow to the particular URL. Note that flows may be associated with explicit URLs only.

To view the flow to an individual URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, click the name of the URL for which you want to see the flow.
   The Allowed URL Properties screen opens.
4. On the menu bar, click **Flows to URL**.
   The Flows to URL screen opens.

Adding a flow to a URL

You can manually create a flow to a URL.

To manually create a flow to a URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, click the name of the URL for which you want to see the flow.
   The Allowed URL Properties screen opens.
4. On the menu bar, click **Flows to URL**.
   The Flows to URL screen opens.
5. Above the Flows to URL list, click the **Create** button.
   The Create a New Flow popup screen opens.
6. In the **Referrer URL** box, select one of the following:
   - **Entry Point**: Specifies that the client can enter the application from this URL
   - **URL Path**: Specifies the URL, if the URL is not an entry point.
7. From the **Protocol** list, select the appropriate protocol.
8. In the **Method** list, select the HTTP method that the URL expects a visitor to use to access the authenticated URL, for example, **GET** or **POST**.
9. In the **Frame Target** box, type the index (0-29, or 99) of the HTML frame in which the URL belongs, if the web application uses frames.

   ***Tip**: If your web application does not use frames, type the value **1**.*
10. If this flow can contain a query string or POST data, check the **Allow QS/PD** box.
11. If you want the system to verify query strings or POST data for this flow, check the **Check QS/PD** box.
12. Click **OK**.
    The popup screen closes, and on the Flows to URL screen, you see the URLs from which the authenticated URL can be accessed.

    ***Tip**: Click a URL in the Flows to URL list to open the Flow Properties screen where you can view or modify the flow's properties.*
13. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring a dynamic flow from a URL

Some web applications contain URLs with dynamic names, for example, the links to a server location for file downloads, where the file name may be unique to each user. You can configure the system to detect these URLs by configuring a dynamic flow.

For a ***dynamic flow***, you configure a regular expression that describes the dynamic name, and associate the flow with the URL. The Application Security Manager then extracts the dynamic URL names from URL responses, for which the dynamic flow is configured.

◆ Note

The URL for which you are configuring a dynamic flow must be a referrer URL.

To configure a dynamic flow

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, click the name of the URL for which you want to see the flow.
   The Allowed URL Properties screen opens.
4. On the menu bar, click **Dynamic Flows From URL**.
   The Dynamic Flows From URL screen opens.
5. Click the **Create** button.
   The Create New Dynamic Flow popup screen opens.
6. In the **Prefix** box, type a fixed substring that appears near the top of the HTML source page before the dynamic URL. It may be a name of a section in combination with HTML tags, for example, **<h3>Flows2URL</h3>**.
7. For the **RegExpValue** setting, type a regular expression that specifies the set of URLs that make up the dynamic flow, for example, a set of archive files.
8. For the **Suffix** setting, type a fixed string that occurs in the referring URL's source code, and is physically located after the reference to the dynamic name URL.
9. Click the **OK** button.
   The popup screen closes, and on the Dynamic Flows from URL screen, you see the dynamic flow extraction properties.
10. To put the security policy changes into effect immediately, In the navigation pane, expand **Application Security** and click **URLs**, and then click the **Apply Policy** button in the editing context area.

Configuring login URLs to prevent forceful browsing

Your web application may contain URLs that should be accessed only through other URLs. For example, in an online banking application, account holders should be able to access their account information only by logging on through a login screen first. In your security policy, you can configure login URLs to limit access to authenticated URLs. A ***login URL*** is a URL in a web application that requests must pass through to get to the authenticated URLs. You can prevent forceful browsing of restricted parts of the web application, by defining access permissions for users.

Using the login pages feature, you can specify one or more login URLs for a web application. If a user tries to bypass the login URLs, the system issues the **Login URL bypassed** violation. You can also configure login page settings that apply to all login URLs including the expiration time, authenticated URLs, and logout URLs. If a user session is idle and exceeds the expiration time, the system issues the **Login URL expired** violation, and the user can no longer reach the authenticated URLs. You can use login URLs to enforce idle time-outs on applications that are missing this functionality.

For both login violations, if the enforcement mode is blocking, the system sends the Login Page Response to the client. For information on response pages, see *Configuring the response pages*, on page 6-46.

To configure login URLs

1. In the navigation pane, expand **Application Security** and click **Flows**.
   The Flows List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. On the menu bar, click **Login Pages**.
   The Login URLs screen opens.
4. Click the **Create** button.
   The New Login URL screen opens.
5. For the **Login URL** setting, select the appropriate protocol, and then type the URL.
6. In the Access Validation area, define at least one validation criteria for the login URL response. See the online help for definitions of these settings.
7. Click the **Create** button to add the login URL to the security policy. The new login URL appears in the Login URLs area.
8. Add as many login URLs as needed for your web application.
9. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

To configure login page settings

1. In the navigation pane, expand **Application Security**, point to **Flows**, point to **Login Pages**, and click **Login Pages Settings**.
2. If you want the login URL to be valid for only a certain length of time, set **Expiration Time** to **Enabled**, and type a value, in seconds.
3. For the **Authenticated URLs** setting, type the name of the URL (may include a wildcard) for which the login URL restricts access, and then click **Add**. Add as many authenticated URLs as needed.
4. Optionally, for the **Logout URL** setting, type the name of the URL that, when accessed, the user must go through the login URL again before being able to access the authenticated URLs. Then click **Add**. Add as many logout URLs as needed.
5. Click the **Save** button.
6. To put the security policy changes into effect immediately, click **Apply Policy**.

Masking sensitive data

Depending on the web application, a response may contain sensitive user information, such as credit card numbers or social security numbers (U.S. only). The Data Guard™ feature can prevent responses from exposing this sensitive information. This process is known as ***response scrubbing***.

In addition to protecting credit card numbers and social security numbers, you can configure custom patterns using PCRE regular expressions to match other types of sensitive information, and indicate which URLs to examine for sensitive data. You can also specify exception patterns that you do not want to be considered sensitive data.

When the system detects sensitive information in a response, and you have enabled the Data Guard feature, the system generates the **Information leakage detected** violation. Additionally, if the security policy enforcement mode is blocking, the system does not send the response to the client.

◆ Note

*When you enable the **Mask Data** option, the system replaces the sensitive data with asterisks (********). F5 Networks recommends that you enable this setting if the security policy enforcement mode is transparent. Otherwise, when the system returns a response, sensitive data could be exposed to the client.*

To configure response scrubbing

1. In the navigation pane, expand **Application Security** and click **Data Guard**.
   The Data Guard screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. If you want the system to consider credit card numbers as sensitive data, check the **Credit Card Numbers** box.
4. If you want the system to consider U.S. social security numbers (in the form **nnn-nn-nnnn**, where **n** is an integer) as sensitive data, check the **U.S. Social Security Numbers** box.
5. To specify additional patterns for sensitive data:
   a) Check the **Enable Custom Patterns** box.
   b) In the **New Pattern** box, type a PCRE regular expression to specify the sensitive data pattern, then click **Add**.
   c) Add as many custom patterns as you need.

6. To specify patterns in the data not to be considered sensitive:
   a) Check the **Enable Exception Patterns** box.
   b) In the **New Pattern** box, type a PCRE regular expression to specify the pattern that you do not want to be considered sensitive (for example, **999-[/d][/d]-[/d][/d][/d][/d]**), then click **Add**.
   c) Add as many exception patterns as you need.
7. If, in the response, you want the system to replace the sensitive data with asterisks (****), check the **Mask Data** box.
8. Use the **Enforcement Mode** setting to specify which URLs to examine for sensitive data:
   - To inspect all URLs, use the default value of **Enforce all URLs**.
   - To check only specific URLs for sensitive data:
     a) Select **Enforce URLs from the list**.
     b) In the **New URL** box, type a URL (explicit or wildcard) and click **Add.**
     c) Repeat step b) to add as many URLs as you need.
9. Click the **Save** button to retain any changes you made.
10. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring allowed modified cookies

By defining allowed modified cookies, you can configure the security policy to ignore certain cookie headers included in an HTTP request, even if they do not meet the expected criteria, and would otherwise trigger a security policy violation. Typically, allowed modified cookies are cookies that can change on the client side, usually using JavaScript™, and these cookies are not session-related.

If you want to use wildcards for allowed modified cookies, refer to *Using wildcards for allowed modified cookie headers*, on page 9-18.

To define an allowed modified cookie

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed Modified Cookies area, click the **Create** button.
   The New Allowed Cookie screen opens.

4. From the **Cookie Name Type** list, select whether the system identifies the cookie by a specific name (**Explicit**), or by a regular expression (**Wildcard**).
5. In the **Cookie Name** box, type either the name of the allowed cookie, or the pattern string for the wildcard to match cookie names.

   ***Tip**: For details on wildcard syntax, refer to **Understanding wildcard syntax**, on page 9-1.*
6. If you want the system to add explicit cookies that match the wildcard cookie, check the **Tightening** box.
7. Click the **Create** button.
   The screen refreshes, and you can see the newly created allowed cookie in the Allowed Modified Cookies list.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Editing an allowed modified cookie

You can edit allowed modified cookies, as required by changes in the web application.

To edit an allowed modified cookie

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Cookie Name column, click the cookie name.
   The Edit Allowed Cookie screen opens.
4. In the Allowed Cookie Properties area, make any needed changes to the cookie.
5. Click the **Update** button.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Deleting an allowed modified cookie

You can delete an allowed modified cookies, as required by changes in the web application.

To delete an allowed cookie

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed Modified Cookies area, click the check box next to an existing allowed cookie name.
4. Click the **Delete** button.
   A confirmation popup screen opens.
5. Click **OK**.
   The system removes the cookie from the security policy.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring mandatory headers

If your application uses custom headers that must occur in every request, you can configure them as mandatory headers in the security policy. A ***mandatory header*** is a header that must appear in a request for the request to be considered legal by the system. If a request does not contain the mandatory header and the **Mandatory HTTP header is missing** violation is set to alarm or block, the system logs or blocks the request. This violation is not set to alarm or block by default, so you have to set the blocking policy if you want to alarm or block requests that do not include a mandatory header.

You can use mandatory headers to make sure, for example, that requests are passing a proxy (which introduces such a header) before they reach the Application Security Manager.

To configure a mandatory header

1. In the navigation pane, expand **Application Security**, point to **Headers** and click **Mandatory Headers**.
   The Mandatory Headers screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Click the **Create** button.
   The New Mandatory Header screen opens.
4. In the **New Header** box, type the name of the header you want to make mandatory. Mandatory headers are not case-sensitive.
5. Click the **Create** button.
   The screen refreshes, and you see the new mandatory header in the list.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring allowed methods

All security policies accept standard HTTP methods by default. The default allowed methods are GET, HEAD, and POST. The system treats any incoming HTTP request that uses an HTTP method other than the allowed methods as an invalid request. If your web application uses HTTP methods other than the default allowed methods, you can add them to the security policy.

To add allowed methods

1. In the navigation pane, expand **Application Security** and click **Methods**.
   The Allowed Methods screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed Methods area, click the **Create** button.
   The New Allowed Method screen opens.
4. For the **Method** setting, choose one of the following actions:
   - Select the system-supplied method that you want to add to the allowed methods list.
   - Select **New Method**, and then type the name of a method in the **New Method** box. Use this option if the method you want to allow is not in the system-supplied list.
5. From the **Act as Method** list, select one of the following options:
   - **GET**: Specifies that the request does not contain any HTTP data following the HTTP header.
   - **POST**: Specifies that the request contains HTTP data following the HTTP header.
6. Click the **Create** button.
   The screen refreshes, and on the Allowed Methods screen, you can see the additional allowed method in the Allowed Methods area.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

To edit or delete existing allowed methods

In addition to creating allowed methods, on the Methods screen, you can also edit or delete allowed methods (except GET and POST), as required by changes in the web application.

- To display the Methods screen, in the navigation pane, expand **Application Security** and click **Methods**.
- To delete an allowed method, check the box next to it, and click **Delete**.
- To edit an allowed method, click the method name to display the method properties, modify as needed, and click **Save**.

Configuring security policy blocking

You can configure when a security policy blocks traffic in several ways:

- **Blocking policy**
  The ***blocking policy*** specifies the blocking actions for each of the security policy violations. The blocking policy also specifies the enforcement mode for the security policy. For more information, see *Configuring the blocking policy*, on page 6-41.
- **Evasion techniques detection**
  Sophisticated hackers have figured out coding methods that normal attack signatures do not detect. These methods are known as ***evasion techniques***. Application Security Manager can detect the evasion techniques, and you can configure blocking properties for them. For more information, see *Configuring blocking properties for evasion techniques*, on page 6-44.
- **HTTP Protocol Compliance**
  The system performs validation checks on HTTP requests to ensure that the requests are formatted properly. You can configure which validation checks are enforced by the security policy. For more information, see *Validating HTTP protocol compliance*, on page 6-14.
- **Web Services Security**
  You can configure which web services security errors must occur for the system to learn, log, or block requests that trigger the errors. For information on how to configure web services security errors, see *Configuring blocking properties for web services security*, on page 6-45.
- **Response pages**
  When the enforcement mode is blocking, and a request (or response) triggers a violation for which the Block action is enabled, the system returns the response page to the client. If you configure login pages, you can also configure a response page for blocked access. For more information, see *Configuring the response pages*, on page 6-46.

Configuring the blocking policy

The Blocking Policy screen is where you configure how the security policy reacts to a request that does not comply with the security policy. On the Blocking Policy screen, you configure the enforcement mode for the security policy, and the blocking actions for all of the violations.

The Blocking Policy screen lists the security policy violations that the system can detect in these categories:

- RFC violations
- Access violations
- Length violations
- Input violations
- Cookie violations
- Negative security violations

Click the information icon ( ) by a violation, or refer to Appendix A, *Security Policy Violations*, for descriptions of the violations. For information on setting the learning, alarm, and blocking actions for the violations, see *Configuring the blocking actions*, on page 6-43.

Configuring the enforcement mode from the blocking configuration

The security policy has two enforcement modes: transparent and blocking. In ***transparent mode***, the system allows requests to reach the web application even if the request violates some aspect of the security policy. In ***blocking mode***, the system does not allow requests that violate the security policy to reach the web application, and instead returns the blocking response page to the client. Note that the system blocks requests only for those violations with enabled Block flags. See *Configuring the blocking actions*, on page 6-43, for more information on the Block flag.

Tip

You can set the enforcement mode from either the Policy Properties screen or the Blocking Policy screen.

To set the enforcement mode from the Blocking Policy screen

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. From the Blocking menu, choose Settings.
   The Blocking Policy screen opens.
3. In the editing context area, ensure that the edited security policy is the one you want to update.
4. In the Configuration area, for the **Enforcement Mode** setting, select either **Transparent** or **Blocking**.
5. Click the **Save** button to save any changes you may have made on this screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring the blocking actions

On the Blocking Policy screen, you can enable or disable the **Learn**, **Alarm**, and **Block** flags, or blocking actions, for violations. The blocking actions determine how the system processes requests that trigger the corresponding violation.

The system takes the following actions when the blocking actions are enabled:

- **Learn**
  When the **Learn** flag is enabled for a violation, and a request triggers the violation, the system logs the request and generates learning suggestions. The system takes this action when the security policy is in either the transparent or blocking enforcement mode.
- **Alarm**
  When the **Alarm** flag is enabled for a violation, and a request triggers the violation, the system logs the request, and also logs a security event. The system takes this action when the security policy is in either the transparent or blocking enforcement mode.
- **Block**
  The **Block** flag blocks traffic when (1) the security policy is in the blocking enforcement mode, (2) a violation occurs, and (3) the **Block** flag is enabled for the violation. The system sends the blocking response page (containing a Support ID to identify the request) to the client.

To customize the blocking actions for security policy violations

1. In the navigation pane, expand **Application Security**, point to **Policy**, point to **Blocking**, then click **Settings**.
   The Blocking Policy screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. Review each violation and adjust the **Learn**, **Alarm**, and **Block** flags as required. Note that you can configure the **Block** flags only when the enforcement mode of the security policy is set to **Blocking**.
4. Click **Save** to save any changes you may have made on this screen.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring blocking properties for evasion techniques

For every HTTP request, Application Security Manager examines the request for evasion techniques, which are coding methods used by attackers designed to avoid detection by attack signatures. You can enable or disable the blocking properties for evasion techniques.

Note

You configure the blocking properties for evasion techniques on the Blocking Policy screen. See ***Configuring the blocking policy****, on page 6-41, for more information.*

To enable blocking properties for evasion techniques

1. In the navigation pane, expand **Application Security**, point to **Policy**, point to **Blocking**, then click **Evasion Techniques**. The Evasion Techniques screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. For each evasion technique, check or clear the **Enable Evasion Technique Checks** check box as required.
4. Click the **Save** button to retain any changes you may have made.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Tip

To return the evasion technique checks to the default settings, click the ***Restore Defaults*** *button.*

Configuring blocking properties for HTTP protocol compliance

You can configure which HTTP protocol compliance checks the security policy validates and enforces. If a request fails any of the enabled HTTP protocol compliance checks, the system responds according to the Learn/Alarm/Block settings of the **HTTP protocol compliance failed** violation on the Blocking Policy screen. For information on configuring the compliance checks, see *Validating HTTP protocol compliance*, on page 6-14.

Configuring blocking properties for web services security

You can configure which web services security errors must occur for the system to learn, log, or block requests that trigger the errors. These errors are sub-violations of the parent violation **Web Services Security failure** configured on the Blocking Policy screen, as described in *Configuring the blocking policy*, on page 6-41.

If you enable any of these errors and one of them occurs, web services security stops parsing the document. How the system reacts depends on how you configured the blocking settings for the **Web Services Security failure** violation:

- If configured to **Learn** or **Alarm** when the violation occurs, the system does not encrypt or decrypt the SOAP message, and sends the original document to the web service.
- If configured to **Block** when the violation occurs, the system blocks the traffic and prevents the document from reaching its intended destination.

To configure web services security errors

1. In the navigation pane, expand **Application Security**, point to **Policy**, point to **Blocking**, then click **Web Services Security**. The Web Services Security errors screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. For each of the web services security sub-violations, check or clear the **Enable** check box as required.
4. Click the **Save** button to retain your changes.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

◆ Tip

*To return the web services security errors to the default settings, click the **Restore Defaults** button.*

Configuring the response pages

The Application Security Manager has a default blocking response page that it returns to the client when the client request, or the web server response, is blocked by the security policy. The system also has a login response page for login violations.

All response pages contain a variable, **<%TS.request.ID()%>**, that the system replaces with a support ID number when the system issues the page. Customers can use the support ID to identify the request when making inquiries.

◆ Note

*The system issues response pages only when the enforcement mode is set to **Blocking**.*

Configuring the blocking response page or the login page response page

The following options are available for the response pages:

- You can use the default response page.
- You can customize a blocking response page.
- You can upload a custom blocking response page.
- You can provide a URL for redirection.
- You can use the default XML (SOAP fault) response page.

To customize a blocking or login response page

1. In the navigation pane, expand **Application Security**, point to **Policy**, and then click **Response Page**.
   The Blocking Response Page screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. Click the **Edit** button for either the blocking or login response page.
   The Response Page Properties screen opens.
4. For the **Response Type** setting, select one of the following options:
   - **Default Response**: Specifies that the system returns the system-supplied response page in HTML. Note that you cannot edit the text.
   - **Custom Response**: Specifies that the system returns a response page with HTML code that you define.
   - **Redirect URL**: Specifies that the system redirects the user to a specific web page.
   - **SOAP Fault**: Specifies that the system returns the system-supplied blocking response page in XML format. Note that you cannot edit the text.

   ***Note**: The settings on the screen change depending on the selection that you make for the **Response Type** setting.*

5. If you selected the **Redirect URL** option in step 4, then in the **Redirect URL** box, type the URL to which the system redirects the user, for example, **http://www.myredirectpage.com**. The URL that you configure should be for a page that is not within the web application itself.

   To redirect the blocking page to a URL with a support ID in the query string, type the URL and the support ID in the following format:

   ```
   http://www.myredirectpage.com/block_pg.php?support_id=
   <%TS.request.ID()%>
   ```

   The system replaces **<%TS.request.ID%>** with the relevant support ID so that the blocked request is redirected to the URL with the relevant support ID.

6. If you selected the **Custom Response** option in step 4, you can either modify the default text or upload an HTML file.

   To modify the default text:

   a) For the **Response Header** setting, click the **Paste Default Response Header** button, and make any changes as required. Use standard HTTP syntax.

   b) For the **Response HTML Code** setting, click the **Paste Default Response HTML Code** button, and make any changes as required. Use standard HTTP syntax.

   To upload an HTML file:

   a) For the **Upload HTML File** setting, either type a path to an HTML response page in the box, or click **Browse** and navigate to an HTML response page.

   b) Click **Upload** when you are finished.

7. Click **Save**.
   The Blocking Response Page opens.

8. For either the blocking or login response page, click **Show.**
   A popup screen shows the text as it will appear to recipients.

9. To put the security policy changes into effect immediately, click the **Apply Policy** button near the top of the Policy Properties screen.

Configuring CSRF protection

Cross-site request forgery (CSRF) is an attack that forces a user to execute unwanted actions on a web application in which the user is currently authenticated.

You can configure a security policy to protect against CSRF attacks, including specifying which URLS you want the system to check. If the system detects a CSRF attack, it issues a **CSRF attack detected** violation. The system inserts an Application Security Manager token to prevent CSRF attacks. To prevent token hijacking, the system checks the token expiration date. If the token is expired, the system issues the **CSRF authentication expired** violation.

If you want to block requests suspected of being CSRF attacks, the security policy enforcement mode must be set to Blocking. Also, one or both of the CSRF violations must have the Block flag enabled (on the Blocking Policy screen), as shown in Figure 6.3.

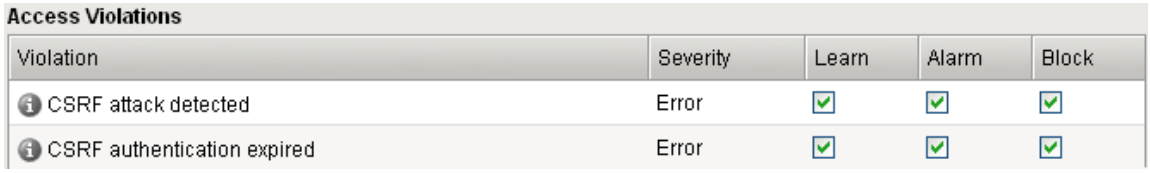


Figure 6.3 *CSRF violations set to block attacks*

To configure CSRF protection

1. In the navigation pane, expand **Application Security**, point to **Policy**, and then click **CSRF Protection**.
   The CSRF Protection screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. Specify which part of the application you want to protect:
   - To protect only SSL requests in the secured part of the application, check the **SSL Only** box.
   - To protect the entire web application, leave the **SSL Only** check box cleared.
4. If you want the CSRF session cookie (which is injected into responses) to expire:
   a) For **Expiration Time**, select **Enabled**.
   b) In the box, type the amount of time, in seconds (**1** to **99999**), after which the cookie should expire.

5. For **URLs List**, select the option that indicates how to use the URLs list when performing CSRF protection:
   - **Enforce only on URLs in the URLs List**
     Specifies that the system considers the URLs in the URLs List unsafe and examines them. The system considers all other URLs safe and does not examine them. This is the default setting.
   - **Enforce on all URLs except those found in the URLs List**
     Specifies that the system considers all URLs unsafe and examines them, except for those URLs in the URLs List which the system considers safe and therefore does not examine.
6. For URL, type an URL that you want to add to the URLs List and click Add. Add as many URLs as you need.

   ***Tip**: You can also use wildcards when defining URLs; some examples are **/myaccount/*.html**, **/*/index.php**, or **/index.?html**.*
7. Click the **Save** button to save your changes.
8. In the navigation pane, point to **Policy**, and then click **Blocking**.
9. For the CSRF violations (**CSRF attack detected** and **CSRF authentication expired**), enable either or both of the **Alarm** and **Block** check boxes. For background details on setting up blocking, refer to *Configuring the blocking policy*, on page 6-41.
   - To block requests suspected of being a CSRF attack, for **CSRF attack detected**, enable the **Block** check box.
   - To block requests containing an expired CSRF session cookie, for **CSRF authentication expired**, enable the **Block** check box.
10. Click **Save** to save the blocking policy.
11. To put CSRF protection into effect immediately, click the **Apply Policy** button in the editing context area.

7

Configuring Anomaly Detection

- What is anomaly detection?
- Preventing DoS attacks for Layer 7 traffic
- Mitigating brute force attacks
- Configuring IP address enforcement
- Detecting and preventing web scraping

What is anomaly detection?

Anomaly detection is a way of detecting patterns in traffic that do not conform with normal behavior, such as an increase in latency or the transactions rate. Application Security Manager™ provides ways for you to configure the system to detect and mitigate anomalies, including:

- **Denial-of-service (DoS) attacks**: Detects the spikes and anomalies in Layer 7 (application layer) traffic.
- **Brute force attacks**: Protects against hackers forceful attempts to gain access to a web site.
- **Increased violations from certain IP addresses**: Prevents attacks originating from specific IP addresses.
- **Bot detection and web scraping**: Prevents automated extraction of data from web sites.

You can add anomaly detection to a security policy developed to protect a web application.

Preventing DoS attacks for Layer 7 traffic

A ***denial-of-service attack*** (DoS attack) is an attempt to make a computer resource unavailable to its intended users. The DoS attack generally consists of the concerted, malevolent efforts to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely. Perpetrators of DoS attacks typically target sites or services, such as banks, credit card payment gateways, and e-commerce web sites.

One common method of attack involves saturating the target (victim) machine with external communications requests, so that the target system cannot respond to legitimate traffic, or responds so slowly as to be rendered effectively unavailable. In general terms, DoS attacks are implemented by forcing the targeted computer to reset, or by consuming its resources so that it can no longer provide its intended service, or by obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately.

Denial-of-service attacks are also known as HTTP-GET attacks or page flood attacks. The attacks can either be initiated from a single user (single IP address) or from thousands of computers (distributed DoS attack), which overwhelms the target system. A page flood attack (or HTTP flood attack) may resemble the patterns of normal Web surfing, making it harder for automated tools to differentiate between legitimate Web traffic and an attempted attack.

Recognizing DoS attacks

Application Security Manager considers traffic to be a DoS attack based on calculations for transaction rate (TPS-based) or latency (latency-based). You can configure which calculations you want the system to use.

If you choose TPS-based, the system detects DoS attacks based on the following calculations:

- **Transaction rate during detection interval**
  The average number of requests per second sent for a specific URL, or by a specific IP address. The system calculates this number every minute.
- **Transaction rate during history interval**
  The average number of requests per second sent for a specific URL, or by a specific IP address. The system calculates this number every hour.

If the ratio of the transaction rate during the detection interval to the transaction rate during the history interval is greater than the specific percentage you configure on the DoS Attack Prevention screen (the **TPS increased by** percentage), the system considers the URL to be under attack, or the IP address to be suspicious. To prevent further attacks, the system drops requests for this URL, and drops requests from the suspicious IP address.

If you choose latency-based, DoS attacks are detected based on the following calculations:

- **Latency during detection interval**
  The average time it takes for the system to respond to a request for a specific URL, for each web application, over the last minute. This average is updated every second.
- **Latency during history interval**
  The average time it takes for the system to respond to a request for a specific URL, for each web application, over the last hour. This average is updated every minute.

If the ratio of the latency during the detection interval to the latency during the history interval is greater than the percentage you configure on the DoS Attack Prevention screen (the **Latency increased by** percentage), the system detects that this URL is under attack.

Configuring DoS attack mitigation

You can configure the Application Security Manager to monitor latency of Layer 7 (application layer) traffic and transactions per second to detect the spikes and anomalies in the typical traffic pattern. The spikes and anomalies may indicate an attempted attack. The system rejects suspicious traffic based on absolute IP addresses and your configuration.

To configure DoS detection and mitigation

1. In the navigation pane, expand **Application Security** and click **Anomaly Detection**.
   The DoS Attack Prevention screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. For **Operation Mode**, select how you want the system to react to DoS attacks:
   - **Off**
     Does not check for DoS attacks. This is the default setting.
   - **Transparent**
     Displays information about DoS attacks on the DoS Attacks reporting screen.
   - **Blocking**
     Displays information about DoS attacks on the DoS Attacks reporting screen, and drops connections coming from an attacking IP address or going to a URL being attacked.

4. For the **Detection Mode**, select the way you want the system to look for DoS attacks:
   - **TPS-based**
     Determines DoS attacks from the client side based on the number of requests per second sent to a specific URL, or the number of transactions per second coming from a specific IP address. This is the default setting.
   - **Latency-based**
     Determines DoS attacks from the server side based on the average time it takes for the system to respond to a request for a specific URL.
5. If you select **Latency-based**, specify the threshold values for **Suspicious Criteria**:
   - **Latency increased by**: Specifies that the system considers traffic to be an attack if the latency has increased by this percentage. The default value is **500**%.
   - **Latency reached**: Specifies that the system considers traffic to be an attack if the latency is equal to or greater than this value. This setting provides an absolute value, so, for example, if an attack increases latency gradually, the increase might not exceed the **Latency Increased by** threshold and would not be detected. If server latency reaches the **Latency reached** value, the system considers traffic to be an attack even if it did not meet the **Latency increased by** criterion. The default value is **10000** ms.
   - **Minimum Latency Threshold for detection**: Specifies that the system considers traffic to be an attack if the detection interval for a specific URL equals, or is greater than, this number, and at least one of the **Latency increased by** number was reached. If the detection interval is lower than this number, the system does not consider this traffic to be an attack even if the **Latency increased by** number was reached. The default setting is **200** ms.
6. For the **Prevention Policy** setting, select one or more options to determine how you want the system to handle a DoS attack:
   - **Source IP-Based Client-Side Integrity Defense**
     Checks whether a client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious IP addresses are requested. Legal browsers can process JavaScript and respond properly, whereas illegal scripts cannot. The default is disabled.
   - **URL-Based Client-Side Integrity Defense**
     Checks whether a client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious URLs are requested. Legal browsers can process JavaScript and respond properly, whereas illegal scripts cannot. This setting enforces strong protection and prevents distributed DoS attacks but affects more clients. The default is disabled.
   - **Source IP-Based Rate Limiting**
     Check to drop requests from suspicious IP addresses. Application Security Manager drops connections to limit the rate of requests

to the average rate prior to the attack, or lower than the absolute threshold specified by the IP detection **TPS reached** setting. The default is enabled.

- **URL-Based Rate Limiting**
  Check to indicate that when the system detects a URL under attack, Application Security Manager drops connections to limit the rate of requests to the URL to the average rate prior to the attack.

7. For **IP Detection Criteria**, type the threshold values:

   ***Note**: This setting appears only if **Prevention Policy** is set to **Source IP-Based Client Side Integrity Defense** and/or **Source IP-Based Rate Limiting**.*

   - **TPS increased by**: Specifies that the system considers an IP address to be that of an attacker, if the transactions (requests) sent per second have increased by this percentage. The default value is **500**%.
   - **TPS reached**: Specifies that the system considers an IP address to be suspicious if the number of transactions (requests) sent per second from an IP address is equal to or greater than this value. This setting provides an absolute value, so, for example, if an attack increases the number of transactions gradually, the increase might not exceed the **TPS increased by** threshold and would not be detected. If the TPS reaches the **TPS reached** value, the system considers traffic to be an attack even if it did not meet the **TPS increased by** criterion. The default value is **200** TPS.

   If either of these criteria is met, the system handles the attack according to the **Prevention Policy** settings.

8. For **URL Detection Criteria**, type the threshold values:

   ***Note**: This setting appears only if **Prevention Policy** is set to **URL-Based Client Side Integrity Defense** and/or **URL-Based Rate Limiting**.*

   - **TPS increased by**: Specifies that the system considers a URL to be an attack if the number of transactions (requests) sent per second to the URL have increased by this percentage. The default value is **500**%.
   - **TPS reached**: Specifies that the system considers a URL to be suspicious if the number of transactions (requests) sent per second to the URL is equal to or greater than this value. This setting provides an absolute value, so, for example, if an attack increases the number of transactions gradually, the increase might not exceed the **TPS Increased by** threshold and would not be detected. If the TPS reaches the **TPS reached** value, the system considers traffic to be an attack even if it did not meet the **TPS increased by** criterion. The default value is **1000** TPS.

   If either of these criteria is met, the system handles the attack according to the **Prevention Policy** settings.

9. For the **Prevention Duration** setting, specify the length of time for which the system mitigates DoS attacks:
   - **Unlimited**: Select if you want the system to perform attack prevention until it detects the end of the attack.
   - **Maximum**: Select and type a value, in seconds. The system prevents detected DoS attacks for the time configured here (even if the attack is still occurring), or until the system detects the end of the attack, whichever is sooner.
10. In **IP Address Whitelist**, type the IP addresses and subnets that do not need to be checked for DoS attacks, and click **Add**.
11. Click **Save** to save the detection and prevention criteria.
12. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

You can view details about DoS attacks that the system detected and logged. For information about the DoS Attacks reports, refer to *Viewing DoS Attacks reports*, on page 15-12. You can also configure remote logging support for DoS attacks when creating a logging profile. For information about creating remote logging profiles, refer to *Configuring a logging profile for remote storage*, on page 14-6.

Mitigating brute force attacks

You can configure the Application Security Manager to detect, report on, and prevent Layer 7 brute force attack attempts. ***Brute force attacks*** are attempts to break in to secured areas of a web application by trying exhaustive, systematic user name/password combinations to discover legitimate authentication credentials. Malicious users send high volumes of these combinations, often scripted, to avoid security mechanisms. In this automated scenario, one malicious user can make thousands of login attempts per minute.

In Application Security Manager, you can configure the login URL of a web application to protect, the mitigation methods, and the access validation criteria for logon responses. With brute force protection, the system can monitor traffic to detect excessive failures to authenticate, monitor suspicious IP addresses (generating a high volume of login attempts), and detect other anomalies in the typical traffic pattern for the login URL.

Application Security Manager tracks the number of failed login attempts for each web application URL in two intervals:

- **History interval**: Failed login attempts for the past hour (updated every minute).
- **Detection interval**: Failed login attempts for the past minute (updated every second).

The system considers it to be a brute force attack if the failed login rate during the detection interval exceeds the failed login rate during the history interval.

To start a new brute force protection configuration

1. In the navigation pane, expand **Application Security**, point to **Anomaly Detection**, and click **Brute Force Attack Prevention**. The Brute Force Attack Prevention screen opens.
2. In the editing context area, verify that the current edited security policy is the one you want to update.
3. Click the **Create** button.
   The New Brute Force Protection Configuration screen opens. Next, you configure the login information.

To configure login information for brute force protection

Continue configuring the settings in the Brute Force Protection Configuration area of the New Brute Force Protection Configuration screen.

1. For **Login URL**:
   - Select **Explicit** (to specify one URL) or **Wildcard** (to specify a pattern that will match multiple URLs).
   - Select the type of traffic the site accepts (**HTTP** or **HTTPS**).
   - Type either a specific URL or a wildcard expression (for example, **Login*.php**).

   The system automatically adds any login URL (explicit or wildcard) that you specify on this screen to the security policy's list of allowed URLs.
2. For **Authentication Type**, select the method by which the application server validates user logon credentials:
   - **HTML Form**
     Specifies that the code of the login URL is in HTML format. This is the default setting.
   - **HTTP Basic Authentication**
     Specifies that the web server uses HTTP basic authentication to authenticate users.
   - **HTTP Digest Authentication**
     Specifies that the web server uses HTTP digest authentication to authenticate users.
   - **NTLM**
     Specifies that the web server uses NT LAN Manager (NTLM) authentication to authenticate users.
3. For **Username Parameter Name**, type the user name parameter included in the code of the HTML form. When the system detects this parameter with the **password** parameter, the system recognizes the request as a login attempt. (Applies only to the **HTML Form** authentication type.)

4. For the **Password Parameter Name** setting, type the password parameter written in the code of the HTML form. When the system detects this parameter with the **username** parameter, the system recognizes that request as a login attempt. (Applies only to the HTML Form authentication type.)
   Next, you can configure session-based mitigation.

To configure session-based brute force protection

Session-based mitigation counts the number of failed login attempts that occur during one session for an IP address. When the system detects a brute force attack, it triggers the **Maximum login attempts are exceeded** violation, and applies the blocking policy.

To configure session-based mitigation, continue configuring the settings in the Session-based Brute Force Protection area of the New Brute Force Protection Configuration screen.

1. Above the Session-based Brute Force Protection area, click the **Blocking Settings** link to open the Blocking Policy screen where you can configure the blocking actions that the system takes when the **Maximum login attempts are exceeded** violation occurs.

   ***Note**: See **Configuring the blocking policy**, on page 6-41, for more information.*

2. For **Login Attempts From The Same Client**, type the number of times a user can attempt to log on before the system blocks the request. The default value is **5**.

3. For **Re-enable Login After**, type the number of seconds the user must wait to log on after they have been blocked.
   Next, you can optionally configure dynamic brute force protection.

   Note that it is not required to configure both dynamic brute force protection and session-based brute force protection.

To configure dynamic brute force protection

Dynamic mitigation detects and mitigates brute force attacks based on statistical analysis of the traffic. You configure dynamic mitigation to determine when the system should consider the login URL to be under attack, and how to react to an attack. The system mitigates attacks when the volume of unsuccessful login attempts is significantly greater than the typical number of failed logins.

To configure dynamic brute force protection, use the settings in the Dynamic Brute Force Protection area of the New Brute Force Protection Configuration screen.

1. For **Operation Mode**, select how the system handles brute force attacks:
   - **Off**
     Does not monitor traffic to detect brute force attacks.
   - **Transparent**
     Issues reporting data only on attacks. Do not drop illegal requests. This is the default setting.
   - **Blocking**
     Drops illegal requests and log reporting data.
2. For the **Detection Criteria** setting, specify when to consider login attempts to be an attack.
   - **Failed Logins Attempts increased by**
     The system considers logon attempts to be an attack if, for all IP addresses tracked, the ratio between the detection interval and the history interval is greater than this number. The default setting is **500** percent.
   - **Failed Login Attempts Rate reached**
     The system considers logon attempts to be an attack if, for all IP addresses tracked, the logon rate reaches this number. The default setting is **1000** logon attempts per second.
   - **Minimum Failed Login Attempts**
     The system considers logon attempts to be an attack if, for all IP addresses tracked, the number of logon attempts is equal to, or greater than, this number. This setting prevents false positive attack detection. The default setting is 100 logon attempts per second.
3. For the **Prevention Policy** setting, select the methods you want the system to use to mitigate an attack (the methods are applied in the order listed).
   - **Source IP-Based Client-Side Integrity Defense**
     Check to determine whether the client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious IP addresses are requested. Legal browsers can process JavaScript and respond properly, whereas illegal scripts cannot. The default is disabled.
   - **URL-Based Client-Side Integrity Defense**
     Check to determine whether the client is a legal browser or an illegal script by injecting JavaScript into responses when suspicious URLs are requested. Legal browsers can process JavaScript and respond properly, whereas illegal scripts cannot. The default is disabled.

- **Source IP-Based Rate Limiting**
  Check to drop requests from suspicious IP addresses. Application Security Manager drops connections to limit the rate of login attempts to the average rate prior to the attack. The default is enabled.
- **URL-Based Rate Limiting**
  Check to indicate that when the system detects a URL under attack, Application Security Manager performs rate limiting and limits the rate of all logon requests to the normal level. The default is enabled.

4. For **Suspicious Criteria (per IP address)**, specify how to identify traffic that may be an attack. If at least one of the criteria is met, the system treats the IP address as an attacker, and prevents the attacker from trying to guess the password. The system also limits the number of login attempts to the normal level.
   - **Failed Login Attempts increased by**
     Type a number. Login attempts from an individual IP address are considered an attack if the number of failed login attempts has increased by this percentage over the normal number of failed logins. The default setting is **500** percent.
   - **Failed Login Attempts Rate reached**
     Type a number. An individual IP address is suspicious if the number of login attempts per second from an IP address is equal to or greater than this number. The default setting is **1000** login attempts per second.
5. For the **Prevention Duration** setting, specify the length of time for which the system mitigates brute force attacks.
   - **Unlimited**
     Specifies that after the system detects and mitigates a brute force attack, it performs attack prevention until it detects the end of the attack.
   - **Maximum**
     Specifies that after the system detects and mitigates a brute force attack, it performs attack prevention either for the time configured here (even if the system detects that the attack continues), or until the system detects the end of the attack, whichever is earlier. Type a value, in seconds, in the box.
6. In **IP Address Whitelist**, add the IP addresses or subnets that do not need to be checked for brute force attacks.
   Next, you can define validation criteria to apply to the response of the login URL.

To configure access validation

Continue configuring the settings in the Access Validation area of the New Brute Force Protection Configuration screen.

For **Access Validation**, define at least one validation criterion for responses of the login URL:

- **A string that should appear in the response**
  Type a string that must appear in the response for the system to detect a successful login attempt; for example, **Successful Login**.
- **A string that should NOT appear in the response**
  Type a string that indicates a failed login attempt; for example, **Authentication failed**.
- **Expected HTTP response status code**
  Type an HTTP response code that is sent when the user successfully logs in; for example, **200**.
- **Expected validation header name and value (for example, Location header)**
  Type a header name and value that is sent when the user successfully logs in.
- **Expected validation domain cookie name**
  Type a defined domain cookie name that is sent when the user successfully logs in.
- **Expected parameter name (added to URI links in the response)**
  Type a parameter that is sent when the user successfully logs in.
  Next, you can finish brute force protection configuration.

To complete brute force protection configuration

1. At the bottom of the New Brute Force Protection Configuration screen, click **Create**.
   The screen refreshes, and you see your protected login URL in the list.
2. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

For how you can view details about brute force attacks that the system detected and logged, refer to the section, *Viewing Brute Force Attack reports*, on page 15-13.

Configuring IP address enforcement

You can configure the IP Enforcer to perform enforcement based on IP address. When the IP Enforcer is in blocking mode, Application Security Manager drops requests for any IP address if more than the maximum number of blocked violations occur in one minute. The system detects an attacker and drops all requests from that IP address for a specified time interval. The IP Enforcer can also operate in transparent mode where it reports IP addresses that exceed the blocked violation threshold but it does not drop the requests. By default, IP address enforcement is off.

◆ Note

*IP address enforcement stops traffic only if the security policy's enforcement mode is **Blocking**, and some violations must have the **Block** flag enabled (on the Blocking Policy screen).*

When the IP Enforcer is configured, you can view IP Enforcer statistics to investigate and release the blocked IP addresses, view dropped requests from that IP address, and examine violations that occurred for each IP address. For details, see *Viewing IP Enforcer statistics*, on page 15-13.

To configure IP address enforcement

1. In the navigation pane, expand **Application Security**, point to **Anomaly Detection** and click **IP Enforcer**.
   The IP Enforcer Configuration screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the IP Enforcer Configuration area, for the **Operation Mode** setting, select the mode for IP enforcement:
   - **Off**
     Does not perform IP-based enforcement. This is the default setting.
   - **Transparent**
     Issues reporting data. In this mode, if the violation threshold is exceeded, the system logs the attack data, but does not prevent access to the application by the offending client.
   - **Blocking**
     Prohibits access by this IP address if more than the maximum number of blocked violations occur in one minute.
4. For the **Violation Threshold** setting, type the maximum number of blocked violations that you want to allow per minute from each IP address. The default value is **10** blocked violations per minute.
5. For the **Prevention Duration** setting, type the number of seconds you want to block (or report) requests. The default value is **60**.
6. Click **Save** to save the IP enforcement configuration.

7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Detecting and preventing web scraping

You can configure Application Security Manager to detect and prevent various bot activities and web scraping on web sites that it is protecting. ***Web scraping*** is a technique for extracting information from web sites typically using automated programs, or ***bots*** (short for web robots).

The Application Security Manager attempts to determine whether a web client source is human. This is how it works:

- **Dropped request**
  If the system cannot check requests for human activity, the request is dropped with no further checking. This action occurs only if the **Block** flag is set for the **Web scraping detected** violation. The system does not drop requests if the security policy is running in transparent mode, or if only the **Learn** or **Alarm** flags are set for the violation.
- **Grace interval**
  The grace interval is how many requests the system reviews while trying to detect whether the client is human. During the grace interval, requests are not blocked or reported. What occurs next depends on whether the system detects human activity:
  - **If the system detects human activity**
    The grace interval ends and the system handles the number of requests specified in the **Safe Interval**, then restarts the grace interval and starts checking again.
  - **If the system does not detect human activity**
    The system issues the **Web Scraping Detected** violation until it reaches the number of requests in the **Unsafe Interval**. If the system is configured to block traffic if that violation occurs, the system blocks requests during this time. In transparent mode or if the violation is set to **Alarm** only, the violation is logged and requests are permitted. After reaching the **Unsafe Interval**, the system restarts the grace interval and starts checking again.

The system can accurately detect human users only when all these conditions exist:

- Clients have JavaScript enabled and support cookies.
- Response caching (the RAM cache and the Web Accelerator cache) is turned off.
- The Block setting for the **Web Scraping Detected** violation is enabled on the Blocking Policy screen.

Preventing web scraping detection on certain addresses

If your environment uses legitimate automated tests, you can create a ***white list*** of IP addresses or address ranges from which to allow access. The system does not perform web scraping detection on these addresses. In addition, the system allows requests from well-known search engines and legal web robots including the following:

- Google (**googlebot.com**)
- Yahoo (**crawl.yahoo.net**)
- MSN (**search.msn.com**)
- ask.com (**ask.com**)
- AOL (using **googlebot.com**)

To configure web scraping detection

1. In the navigation pane, expand **Application Security**, point to **Anomaly Detection,** then click **Web Scraping**.
   The Web Scraping screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. For **Grace Interval**, type the number of requests to allow while determining whether the client is human. The default value is **100**.
4. For **Unsafe Interval**, type the number of requests that cause the **Web Scraping Detected** violation if no human activity is detected during the grace period. When the number is reached, reactivate the grace period. The default value is **100**.
5. For **Safe Interval**, type the number of requests to allow after human activity is detected and before reactivating the grace threshold to check again for non-human clients. The default value is **2000**.
6. In **IP Address Whitelist**, add the IP addresses or subnets that do not need to be checked for web scraping.
7. Click **Save**.

You can view details about web scraping attacks that the system detected and logged, as described in *Viewing web scraping statistics*, on page 15-14.

8

Maintaining Security Policies

- Maintaining a security policy
- Reviewing a log of all security policy changes
- Displaying security policies in a tree view
- Using the security policy audit tools

Maintaining a security policy

Security policies can change and evolve over time. As the nature of the traffic through the web application changes, you can adjust the security policy as required. From the Policies List screen, you can perform the following policy maintenance tasks:

- Edit a security policy
- Copy a security policy
- Export a security policy
- Import a security policy
- Merge two security policies
- Remove a security policy from the configuration
- Restore a deleted security policy
- Delete a security policy permanently
- View the history of a security policy
- Restore a previous version of the security policy

You can review all changes that have been made to a security policy by reviewing the policy log.

You can also display a tree view of the security policy to quickly view its contents. For more information on the tree view, refer to *Displaying security policies in a tree view*, on page 8-10.

Editing an existing security policy

You can access a security policy for editing from either the Policies List screen, or from the editing context area. The editing context area appears at the top of almost every screen throughout the Application Security Manager™. Figure 8.1 displays the editing context area.

Figure 8.1 *Editing context area*

To edit a security policy from the Policies List screen

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.

   Note: *If a security policy's entire row is highlighted in gray, this indicates that another user is currently editing it. As a result, you can view but not edit that security policy.*

2. In the Security Policies area, click the name of the security policy that you want to edit.
   The Policy Properties screen opens.
3. Make any changes that are required.
4. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Copying a security policy

You can copy a security policy to quickly duplicate policies or create policies that differ only in a few details.

To copy a security policy

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. In the Security Policies area, select the security policy that you want to copy by clicking the button on its left.
3. Click the **Copy** button.
   The Copy Security Policy screen opens.
4. In the **New Security Policy Name** box, accept or change the name for the security policy, and then click **Save**. The default name is the **<original_policy_name>_copy**.)
   The system displays a message when the policy is successfully copied.
5. Click **OK**.
   The screen refreshes, and you see the new security policy in the Security Policies List.

Note

In the Security Policies List, the Active icon *next to a security policy indicates that this policy is active. The Modified icon* *indicates that the security policy has been modified, and you must click the* ***Apply Policy*** *button to implement any changes in the security policy.*

Exporting a security policy

You can export security policies as a binary archive file or as a readable XML file. For example, you may want to export a security policy from one web application so that you can use it as a baseline for a new web application. You can also export a security policy to archive it on a remote system before upgrading the system software, to create a backup copy, or to use the exported security policy in a policy merge. (See *Merging two security policies*, on page 8-5, for more information on merging policies.)

You can export the security policy on a remote system or other location. The XML or archive file includes the name of the security policy and the date it was exported. If you saved the policy as an XML file, you can open it to view the configured settings of the security policy in a human readable format.

To export a security policy

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. In the Security Policies list, select the security policy that you want to export by clicking the button on its left, then:
   - To save the security policy as a policy archive file (**.plc** file), click **Export**.
   - To save the security policy as an XML file, click **Export as XML**.
3. In the file download screen, save the file.
   The system exports the security policy in the format you specified and saves it in the remote location.

Importing a security policy

You can import a security policy previously saved in archive policy or XML format to quickly apply a security policy to a new web application. You can also use the import option to restore a security policy from a remote system.

To import a security policy

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. Above the Security Policies area, click the **Import** button.
   The Import Security Policy screen opens.
3. In the **Choose File** setting, click the **Browse** button to navigate to the security policy that you want to import.
4. Click **Import**.
   The system displays a success status message when the operation is complete.
5. Click **OK**.
   The screen refreshes, and you can see the imported security policy in the Security Policies List.

◆ Note

The names of security policies must be unique within the Application Security Manager. If the name of the imported security policy already exists, the system renames the imported file by adding a sequential number to the end of the name.

Merging two security policies

You can use the policy merge option to combine two security policies. For example, you can use the policy merge option to merge a security policy that you have built offline into a security policy that is on a production system.

The merge mechanism is lenient when merging security policies. The merge action does not delete anything from the target security policy. The system resolves any conflicts that occur by retaining the settings of the target security policy. When the merge is complete, the system displays the beginning of a merge report of all security policy components that were modified or added during the merge process. In addition, you have the option to view or download the complete merge report. You can save the Policy Merge Report as a text file (***.txt**), so that you can review the details of the merge, and resolve any errors that may have occurred.

◆ Note

When a security policy contains restrictive components, for example, a user-defined attack signature set, the merge tool deletes it.

The merge report contains information about any conflicts that occurred during the merge, and how they were resolved. If you enable verbose logging for the merge, the merge report also contains the following information:

- Entities that are in the target security policy only
- Entities in the target security policy whose values are different from those in the merged security policy
  (If this occurs, the system does not change the target security values.)

To merge two security policies

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. In the Security Policies area, select the target security policy (the one into which the system merges the second security policy) by clicking the button on its left, and click the **Merge** button.
   The Merge Security Policies screen opens.
3. For the **Security Policy To Be Merged** setting, click the **Browse** button, and navigate to the exported security policy file that you want to merge into the target security policy.
4. To save a copy of the original security policy, check the **Backup Target Security Policy** setting.
5. To include additional details about the merge, check the **Verbose Mode** setting.
6. Click the **Merge** button.
   The system merges the export security policy into the target security policy, and produces a Merge Report.

7. Click the **Download Full Report** button to open or save the entire Merge Report.
8. Click **OK**.
   The screen refreshes, and the merged security policy is in the Security Policies list.

   ***Note**: A copy of the original security policy also appears in the Security Policies list, if you selected the **Backup Target Security Policy** option in step 4.*

Removing a security policy from the configuration

You can remove all security policies from the configuration, one by one, except the active security policy. The active security policy for a web application has the Active icon next to its name in the Security Policies list.

To remove a security policy from the configuration

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. In the Security Policies area, select the security policy that you want to remove from the configuration, and click the **Delete** button below the list.
   A confirmation popup screen opens, where you confirm that you want to delete the security policy.
3. Click **OK**.
   The screen refreshes and you no longer see the security policy in the Security Policies List.

Restoring a deleted security policy

If you delete a security policy, and later decide that you did not want to do that, you can restore the security policy from the Security Policy Recycle Bin.

To restore a security policy

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. Above the Security Policies area, click the **Import** button.
   The Import Security Policy screen opens.
3. In the Security Policy Recycle Bin list, select the security policy that you want to restore, and then click the **Restore** button.
   A confirmation popup screen opens, where you confirm that you want to restore the security policy.
4. Click **OK**.
   The system restores the security policy, and displays a success message.
5. Click **OK**.
   The screen refreshes, and you see the restored security policy in the Policies List.

Deleting a security policy permanently

If you delete a security policy from the configuration, and later decide that you want to delete it permanently, you can delete the security policy from the Security Policy Recycle Bin.

To delete a security policy permanently

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. Below the Security Policies area, click the **Import** button.
   The Import Security Policy screen opens.
3. In the Security Policy Recycle Bin list, select the security policy that you want to delete, and then click the **Delete** button.
   A confirmation popup screen opens, where you can confirm that you want to delete the security policy.
4. Click **OK**.
   The screen refreshes, and you no longer see the security policy in the Security Policy Recycle Bin list.

Viewing and restoring an archived security policy

The Application Security Manager keeps an archive of security policies that have been set to active. Every time you make a security policy the active security policy, the system saves a version of that security policy, and archives it. You can restore any of the archived security policies, and make it the active security policy.

Tip

In the Security Policies list, on the Policies List screen, the security policy version number is in square brackets next to the security policy name.

To view a list of the versions of a security policy and restore an archived security policy

1. In the navigation pane, expand **Application Security** and click **Policies List**.
   The Policies List screen opens.
2. In the Security Policies list, click the security policy whose different versions you want to view or whose archived version you want to restore.
   The Policy Properties screen opens.
3. On the menu bar, click **History**.
   The Security Policy History screen opens, where you can view the archived versions of the security policy.
4. To restore an archived security policy, select the version, and then click the **Restore** button.
   The Restore Security Policy screen opens.
5. In the **Security Policy Name** box, change the name as required.
6. If you do not want the restored security policy to be immediately active, clear the **Apply Policy** box.
7. Click **OK**.
   The screen refreshes and you see the restored security policy in the Policies List.

Reviewing a log of all security policy changes

The Application Security Manager creates a policy log for every security policy. The ***policy log*** includes an entry for each event or action performed on the security policy, including the event type, the element type and name (if relevant), the data and time of the change, a description of the change, and where and how the change was made.

This log is different from the automatic policy building log because this one shows all changes that the Policy Builder or a user made to the security policy. The automatic policy building log is described in *Viewing automatic policy building logs*, on page 5-24.

To review all security policy changes

1. In the navigation pane, expand **Application Security**, point to **Policy**, then click **Policy Log**.
   The Policy Log screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those for which you want to view log transactions.
3. In the **Filter** area, adjust the filter settings to view the logs you want to see.
4. Click the **Go** button.
   The screen refreshes, and displays the policy log for the web application and security policy that you selected. Figure 8.2 shows a portion of a sample policy log.
5. In the Description column, click the + magnifying glass to view more details about the change.

Event Type	Element Type	Element Name	Event Time	Description	Component	User
Update	Apply Policy	N/A	2010-03-19 22:06:42	N/A	GUI	admin
Update	URL	*	2010-03-19 22:06:34	Perform Tightening was set to enabled. Updating wildcard in orde...	Policy Builder	N/A
Update	URL	*	2010-03-19 22:06:34	Perform Tightening was set to enabled. Updating wildcard in orde...	Policy Builder	N/A
Update	Policy Builder Configuration	Policy Builder Config	2010-03-19 22:06:34	URLs was set to enabled.	GUI	admin
Update	Policy Builder Configuration	Policy Builder Config	2010-03-19 22:06:21	Tighten Rule (Trusted Traffic): total requests was set to 1000.	GUI	admin

Figure 8.2 *Sample policy log showing all changes to the security policy*

Displaying security policies in a tree view

You can display a tree view of the security policy to quickly view its contents. The tree view shows the complete hierarchy of the web application as reflected in the security policy. Global parameters appear at the top level, and URL parameters fall under URLs in the directory-like structure.

To display a tree view of a security policy

1. In the navigation pane, expand **Application Security**, point to **Policy**, and then click **Tree View**.
2. In the editing context area, ensure that the edited security policy is the one you want to display in tree view.
3. Click a directory (with a plus sign) to expand the directory and view the child elements.
4. Click an allowed URL, a disallowed URL or a parameter to view its properties.
   The properties page for the URL or parameter opens.

Figure 8.3 shows the structure of a security policy for **www.paycom.com**, a web application for selling merchandise.

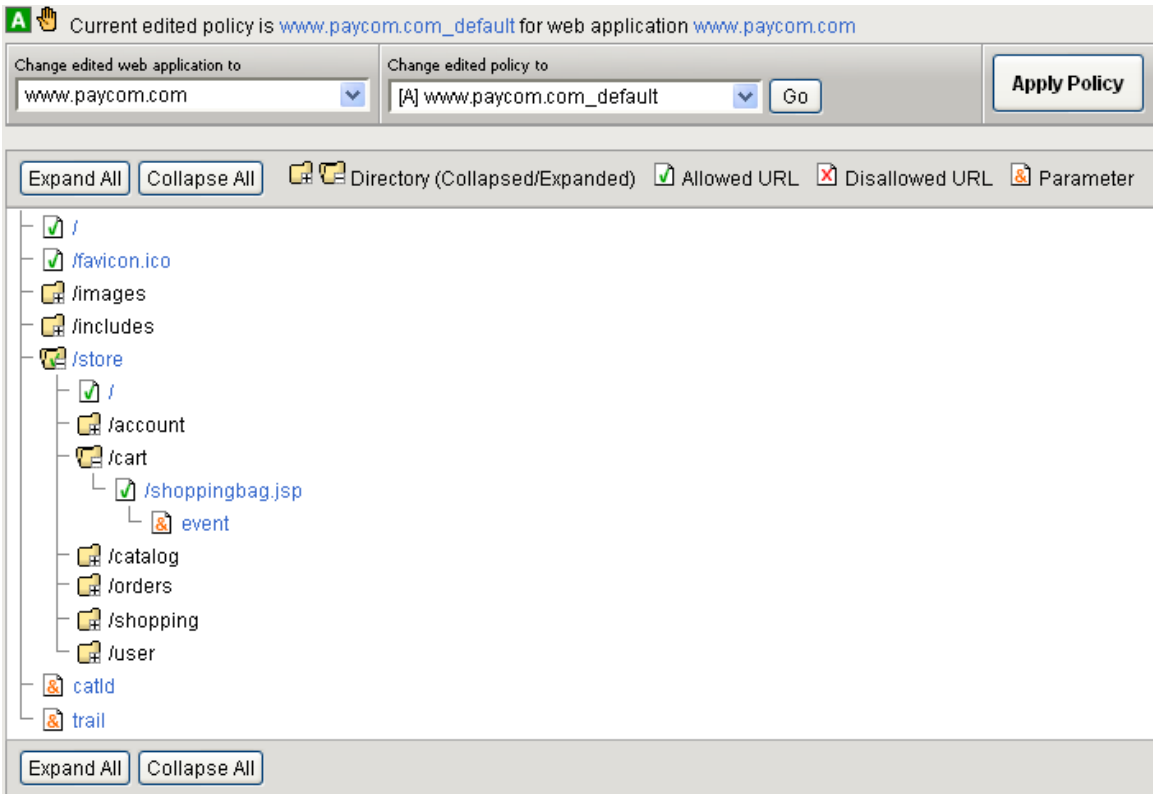


Figure 8.3 *Example tree view of a security policy*

Using the security policy audit tools

Application Security Manager includes several audit tools that you can use to query a security policy to find the information you are looking for. You can use the audit tools to analyze suspicious policy states (for example, URLs allowed to modify domain cookies). Each tool type specifies a predefined URL, parameter, or flow filter that helps to identify conflicts and errors in the security policy.

To use the security policy audit filters

1. In the navigation pane, expand **Application Security**, point to **Policy** and click **Audits**.
   The Policy Audits screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. From the **Tool Type** list, select an audit tool, and then click **Go**.
   The screen refreshes, and the system displays the audit report.

9

Working with Wildcard Entities

- Overview of wildcard entities
- Configuring wildcard file types
- Configuring wildcard URLs
- Configuring wildcard parameters
- Using wildcards for allowed modified cookie headers

Overview of wildcard entities

Wildcard entities are web application entities in the security policy that contain one or more shell-style wildcard characters. In Application Security Manager™, wildcard entities can represent file types, URLs, parameters, and cookie headers. Wildcards provide flexibility for security policy building. By using wildcard entities, you can efficiently build a security policy without in-depth knowledge of the web application, and reduce the number of violations and false-positives.

This chapter describes how to manually create wildcard entities. If you are using automatic policy building, the Policy Builder creates wildcards, adds explicit entities, and when the security policy is stable, removes wildcards and enforces the explicit entities, depending on the configuration. In that case, you do not need to create wildcards because the security policy is built automatically.

Understanding wildcard syntax

The syntax for wildcard entities is based on shell-style wildcard characters. Table 9.1 lists the wildcard characters that you can use in a wildcard entity name.

Wildcard Character	Description
*	Match all characters
?	Match any single character
[seq]	Match any character that is in the specified sequence
[!seq]	Match any character that is not in the specified sequence

Table 9.1 *Wildcard characters and usage descriptions*

The easiest wildcard to configure is the asterisk (*), which the system interprets as match everything. You can use the * character on its own, or in a name.

◆ Note

*If you add to the security policy a wildcard URL that does not begin with the asterisk (*****) character (for example **a*b**), the system does not automatically add the slash (/) character before it. You must manually add the slash (/) character before this type of URL for the system to enforce it.*

Understanding staging and tightening for wildcard entities

When you create a wildcard entity, you have the option to enable staging and tightening for that entity. You can use tightening first to learn explicit entities, review and enforce learning suggestions; the system automatically enables staging for entities you have learned. F5 Networks recommends against using both staging and tightening on the same entity.

Tightening is using wildcards to learn the entities (file types, URLs, parameters, and cookies). ***Staging*** is learning the attributes of an entity (wildcard or explicit), providing additional granularity over tightening.

Understanding tightening

You can perform tightening on wildcard entities (file types, URLs, parameters, and cookies) to learn explicit entities. When you enable tightening for a wildcard entity, and the system receives a request that contains an entity that matches the wildcard entity, the system generates a learning suggestion for the found entity. You can then review the new entities, and decide which are legitimate entities for the web application.

Tightening gives you the option of developing a more specific policy, a policy that is more accurate and in alignment with the traffic. Such a policy can provide better security, but requires more tuning to make sure all the specific entities that you add are accurately configured.

If the Policy Builder is running and the traffic source is trusted (either by definition or because of heuristic decisions), the Policy Builder automatically adds the new specific entity to the security policy.

Note

When you accept learning suggestions, you add explicit entities to the security policy. The next time the system receives a request with that entity, the Security Enforcer applies the security policy to the explicit entry, and not to its parent wildcard entity. Note also that accepting many explicit entities may complicate security-policy maintenance.

Each security policy can have wildcards for file types, URLs, parameters, and cookies. When you create a security policy using the Deployment wizard, the system enables tightening on wildcard entities (depending on the scenario you select). As traffic is sent to the web application, the system learns the explicit properties of the file types, URLs, parameters, and cookies.

Tip

*Use tightening on wildcard entities to build the security policy with explicit entities, and then enforce the entities that are ready to be enforced by using the **Enforce** and **Enforce Ready** buttons. When you accept tightening suggestions for a wildcard, the system automatically places the explicit entity into staging.*

Understanding staging

You can perform staging on wildcard entities (file types, URLs, and parameters) to learn the properties of the entities, as described in Table 9.2.

Wildcard entity	Properties learned in staging
File type	File type lengths (URL length, request length, query string length, or POST data length)
URL	Meta characters
Parameter	Parameter settings

Table 9.2 *Properties learned in staging for wildcard entities*

When an entity is in staging, the system does not block any requests for this entity. Instead, it posts learning suggestions for staged entities on the Learning screens. After the staging period is over and you see that requests for this entity do not log additional learning suggestions, F5 Networks recommends you take the entity out of staging by clearing the **Perform Staging** check box on the file types, URLs, or parameters properties screen. This is necessary only if you are manually building a security policy, and not using automatic policy building.

◆ Tip

Use staging on wildcard entities to build the security policy without explicit entities of this type, so that the wildcard entity itself is enforced with the settings found on it.

Staging is also extremely useful when a site update occurs for a web application. With staging, you can add new URLs or parameters to the security policy and stage only the new entities. You can keep existing policy entities in blocking mode, while placing the new entities in transparent mode, which can generate learning alerts.

To enforce file types, URLs, and parameters that are ready to be enforced

1. In the navigation pane, expand **Application Security**, point to **Manual Policy Building** and click **Staging-Tightening Summary**. The Staging-Tightening Summary screen opens.
2. In the Staging-Tightening Summary, check to see if a number other than **0** appears in the **Ready To Be Enforced** column.
3. Select an entity type that has instances that are ready to be enforced.
4. Click the **Enforce Ready** button.
   A confirmation popup screen opens where you can confirm that you want to enforce all entities that are ready to be enforced for the selected entity types.

5. Click **OK**.
   The screen refreshes; the system performs the following on selected entities:
   - Removes from staging entities whose staging period is over.
   - Deletes wildcard entities whose tightening period is over.
   - Changes the values in the Staging-Tightening Summary columns to **0**.

To enforce file types, URLs, and parameters in staging or with tightening enabled

1. In the navigation pane, expand **Application Security**, point to **Manual Policy Building** and click **Staging-Tightening Summary**.
   The Staging-Tightening Summary screen opens.
2. In the Staging-Tightening Summary, check to see if a number appears in the **In Staging-Tightening** column.
   A number greater than zero indicates that entities of that type were discovered while in staging or with tightening enabled.
3. Click the number to view the file types, URLs, or parameters in staging or with tightening enabled.
   The allowed file types, URLs, or parameters list opens.
4. Select the entities you want to enforce.
5. Click the **Enforce** button.
   A confirmation popup screen opens, where you confirm that you want to enforce all selected entities.
6. Click **OK**.
   The screen refreshes; the system performs the following on selected entities:
   - Removes selected entities (explicit and wildcard) from staging.
   - Deletes from the security policy selected wildcard entities with tightening enabled.

Understanding security policy enforcement for wildcard entities

The Security Enforcer applies the following logic when a security policy includes wildcard entities.

- **Check for explicit matches**
  First, the Security Enforcer checks for an explicit match, that is, the Security Enforcer scans the security policy to verify whether it contains the exact entity. If the security policy contains an explicit matching entity, the system applies the checks that are specified for that entity.
- **Check for wildcard matches**
  If the security policy does not contain an explicit matching entity, the system checks the wildcard entities to determine whether any of them match the requested entity. If the system finds a wildcard match, the

Security Enforcer applies any applicable security checks. If you have enabled tightening for the wildcard entity, the Security Enforcer generates a learning suggestion for the new entity, which the system displays on the Traffic Learning screen.

If the Security Enforcer does not find an explicit match or a wildcard match, the system generates a violation for the illegal entity. If the triggered violation is in blocking mode, the system drops the request and sends the Blocking Response page to the client.

Configuring wildcard file types

File types represent the file type extensions of the files that make up the web application, such as **htm**, **jsp**, and **gif**. You can specify wildcard file types in a security policy when you do not want the overhead of maintaining a list of explicit file types.

For general information on file types, see *Adding file types*, on page 6-16.

Creating wildcard file types

You can create a wildcard file type so that requests do not generate violations based on the requested file type. Any file type that matches the wildcard expression is considered legal. For example, the wildcard * specifies that any file type is allowed by the security policy. By default, allowed file types you create are put into staging. If you want to enable tightening (first disable staging), you can learn which file types are used in the protected web application. For more information, see *Working with entities in staging or with tightening enabled*, on page 13-9.

To create a wildcard file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed File Types list, click the **Create** button.
   The Add Allowed File Type screen opens.
4. For the **File Type** setting, select **Wildcard** from the list, and then type a wildcard string in the box (for example, *****).
5. If you want the system to display explicit file types that match the wildcard entity pattern that you specify, clear the **Perform Staging** check box, and then check the **Perform Tightening** setting.

   ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*

6. Modify the length settings as required.
7. If you want the system to parse responses in addition to parsing requests, check the **Check Response** box.
8. Click the **Create** button to add the wildcard file type to the security policy.
   The screen displays the updated Allowed File Types screen.
9. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Modifying wildcard file types

You can modify the settings for an existing wildcard file type.

To modify a wildcard file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed File Types list, in the Type column, click the name of the file type that you want to modify.
   The Allowed File Type Properties screen opens.
4. Make the necessary adjustments to the settings.
5. Click the **Update** button to update the security policy with any changes you may have made.
   The screen refreshes, and displays the Allowed File Types screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Deleting wildcard file types

You can delete wildcard file types once the explicit file types used in the application have been added to the security policy.

To delete a wildcard file type

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Select column (far left), check the box next to the wildcard file type that you want to remove, and then click the **Delete** button.
   The system displays a confirmation popup screen.
4. Click **OK**.
   The system deletes the wildcard file type from the configuration.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Sorting wildcard file types

When you have configured more than one wildcard file type, you can set the ***enforcement order***, which is the sequence in which the Security Enforcer searches for a match within those wildcard file types. If it finds a match, the Security Enforcer then applies the security checks that are associated with that wildcard entity to the matching entity.

You can organize wildcard file types in the sequence that you want them to be enforced, which is from most-specific to least-specific. The system enforces wildcard file types from the top down.

To set the enforcement order for wildcard file types

1. In the navigation pane, expand **Application Security** and click **File Types**.
   The Allowed File Types screen opens.
2. On the menu bar, click **Wildcards Order**.
   The Wildcards Order screen opens.
3. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
4. For the **Wildcard File Types Lis**t setting, make any adjustment to the list order by using the **Up** and **Down** buttons.
5. Click the **Save** button to save any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring wildcard URLs

URLs represent the pages and images that compose the web application. Wildcard URLs use wildcard characters in the URL name. When you are building a security policy, using wildcard URLs reduces the number of false-positives. You can also use wildcard URLs in a security policy when you do not want the overhead of maintaining explicit URLs. By using wildcard URLs, you can configure security checks for a set of URLs, rather than configuring the security checks for each individual URL.

Note

*For general information on working with URLs, see **Configuring URLs**, on page 6-21.*

Creating wildcard URLs

You can create a wildcard URL so that requests do not generate violations based on the requested URL. Any URL that matches the wildcard expression is considered legal. For example, the wildcard expression * specifies that any URL is allowed by the security policy. By default, allowed wildcard URLs you create are put into staging. If you want to enable tightening (first disable staging), you can learn which URLs are used in the protected web application.

To create a wildcard URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed URLs List area, click the **Create** button.
   The New Allowed URL screen opens.
4. For the **URL** setting, select **Wildcard** from the list, and then type a wildcard string in the box (for example, *****).
   The screen refreshes and displays additional settings for meta characters.
5. If you want the system to display explicit URLs that match the wildcard entity pattern that you specify, clear the **Perform Staging** check box, and then check the **Perform Tightening** setting.

   ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*
6. For the **Protocol** setting, select the web application's protocol.

7. If you want the system to validate XML data in requests to this URL based on the settings configured in an XML profile, check the **Apply XML Profile** setting.

   a) If you already have an XML profile, select one from the list. If not, click the + button to create one for the security policy. For details, see Chapter 12, *Protecting XML Applications*.

   b) For the **Check XML Content-Type Headers** setting, specify how the system applies the XML profile to requests for this URL.

   - Select **All** if you want the system to inspect all requests.
   - Select **User-defined** and type a string if you want the system to inspect only those requests whose **Content-Type** header value contains the string you specified. The default value is ***xml***.

8. If your application uses Action Message Format for content-type headers:

   a) Above the Create New Allowed URL area, select **Advanced**.

   b) Check the **Check AMF (When the content type matches "amf")** box.

9. For the **URL Description** setting, type an optional description.

10. In the Meta Characters area, the **Check characters on this URL** setting is enabled by default so that the system verifies meta characters in the URL. (If you do not want to check for meta characters, clear the check box, and proceed to step 11.)

    Specify which meta characters to allow or disallow:

    a) From the **Global Security Policy Settings** list, select any meta characters that you want to specifically allow or disallow, and move them to the Overridden Security Policy Settings list.

    b) Set the state of each meta character you moved to **Allow** or **Disallow**.

    ***Note**: The Overridden Security Policy Settings take precedence over the global settings for the web application's character set.*

11. Click the **Create** button to add the wildcard URL to the security policy.
    The screen displays the updated Allowed URLs List screen.

12. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
    The system applies the updated security policy.

    ***Tip:** If you enabled staging or tightening and Policy Builder is enabled, the system analyzes traffic going to the web application and adds entities or their properties to the policy. If you did not, you can accept learning suggestions manually. For details, see **Working with entities in staging or with tightening enabled**, on page 13-9.*

Modifying wildcard URLs

At times, you may want to modify the settings for an existing wildcard URL.

To modify a wildcard URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, in the URL column, click the name of the URL that you want to modify.
   The Allowed URL Properties screen opens.
4. Make the necessary adjustments to the settings.
5. Click the **Update** button to update the security policy with any changes you may have made.
   The screen refreshes, and displays the Allowed URLs List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Deleting wildcard URLs

You can easily delete any wildcard URLs that are no longer necessary in the security policy.

To delete a wildcard URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed URLs List area, check the box next to the wildcard URL that you want to remove, and then click the **Delete** button.
   The system displays a confirmation popup screen.
4. Click **OK**.
   The system deletes the wildcard URL from the configuration.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Sorting wildcard URLs

When you have configured more than one wildcard URL, you can set the ***enforcement order***, which is the order in which the Security Enforcer searches for a match within those wildcard URLs. If the Security Enforcer finds a match in the wildcard URLs, the Security Enforcer then applies the security checks that are associated with that wildcard entity to the matching entity.

Tip

When ordering wildcard URLs, you should arrange them in the order in which you want them to be enforced. The system enforces them from the top down.

To set the enforcement order for wildcard URLs

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. On the menu bar, click **Wildcards Order**.
   The Wildcards Order screen opens.
3. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
4. In the Wildcards Order area, for the **Wildcard URLs List** setting, make any adjustment to the list order by using the **Up** and **Down** buttons.
5. Click the **Save** button to save any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring wildcard parameters

You can specify wildcard parameters in a security policy when you do not want the overhead of maintaining a list of explicit parameters. Using wildcard parameters reduces the number of **Illegal parameter** violations you receive when creating a security policy.

If you create a wildcard parameter and enable staging or tightening, you can also learn the types or attributes of the parameter, and which parameters are used in the protected web application.

Note

For more information on working with parameters, see Chapter 10, ***Working with Parameters****.*

Creating wildcard parameters

You create wildcard parameters similarly to the way that you create explicit parameters, with the following exceptions:

- A wildcard parameter cannot be a dynamic content value (DCV) parameter.
- A wildcard parameter cannot have a dynamic parameter name.
- A wildcard parameter cannot be a navigation parameter.

For wildcard parameters that you create, any parameter name that matches the wildcard expression is permitted by the security policy. For example, typing the wildcard * specifies that the security policy allows every parameter. By default, new parameters you create are put into staging. If you want to enable tightening (first disable staging), you can learn which parameters are used in the protected web application.

To create a wildcard parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Parameters List area, click the **Create** button.
   The Add Parameter screen opens.
4. In the Create New Parameter area, for the **Parameter Name** setting, select **Wildcard** from the list, and then type a wildcard string in the box (for example, *****).

5. For the **Parameter Level** setting, select the appropriate option for this wildcard parameter.
   - **Global Parameter**: For more information, see *Working with global parameters*, on page 10-2.
   - **URL Parameter**: For more information, see *Working with URL parameters*, on page 10-5.
   - **Flow Parameter**: For more information, see *Working with flow parameters*, on page 10-8.

   The screen refreshes to display additional settings, depending on the parameter level that you select.
6. If you want the system to display explicit parameters that match the wildcard entity pattern that you specify, clear the **Perform Staging** box, and then check the **Perform Tightening** box.

   ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*
7. To allow requests to contain multiple parameters with the same name, check the **Allow Repeated Occurrences** box. The default setting is disabled.
8. If you want to treat the parameter you are creating as a sensitive parameter (not visible in logs or the user interface), check **Sensitive Parameter**.
9. For the **Parameter Value Type** setting, select the appropriate type from the list.
   The screen refreshes to display additional settings that are relevant to the parameter value type that you selected.

   ***Note**: For detailed information regarding the parameter value type options, see **Understanding parameter value types**, on page 10-12.*
10. Configure the remaining settings as required, and then click the **Create** button.
    The screen refreshes, and displays the new wildcard parameter.
11. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
    The system applies the updated security policy.

◆ Tip

*If you enabled staging or tightening and Policy Builder is enabled, the system analyzes traffic going to the web application and adds entities or their properties to the policy. Otherwise, you can accept learning suggestions manually. For details, see **Working with entities in staging or with tightening enabled**, on page 13-9.*

Modifying wildcard parameters

You may want to modify the settings for an existing wildcard parameter. You can change the properties of a parameter, but you cannot change its name.

For detailed information about the parameter properties, refer to Chapter 10, *Working with Parameters*.

To modify a wildcard parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the wildcard parameter that you want to modify.
   The [Global/URL/Flow] Parameter Properties screen opens.
4. Make the necessary adjustments to the settings.
5. Click the **Update** button to update the security policy with any changes you may have made.
   The screen refreshes, and displays the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.
   A confirmation popup screen opens.
7. Click **OK**.
   The system applies the updated security policy.

Deleting wildcard parameters

You can delete any wildcard parameters that are no longer necessary in the security policy.

To delete a wildcard parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Parameters List area, in the Select column (far left), check the box next to the wildcard parameter that you want to remove, and then click the **Delete** button.
   The system displays a confirmation popup screen.
4. Click **OK**.
   The system deletes the wildcard parameter from the configuration.

5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Ordering wildcard parameters

When you configure more than one wildcard parameter, you can set the ***enforcement order***, which is the sequence in which the Security Enforcer searches for a match within those wildcard parameters. If the Security Enforcer finds a match in the wildcard parameters, the Security Enforcer then applies the security checks that are associated with that wildcard entity to the matching entity. For wildcard parameters, the system looks for matches in this order: flow parameters, URL parameters, then global parameters.

The Security Enforcer always looks for a match on an explicit parameter first. If the explicit parameter is not found, the Security Enforcer looks for the next possible wildcard match on the current level, that is, flow, URL, or global. This process continues for each parameter level, as shown in Figure 9.1.

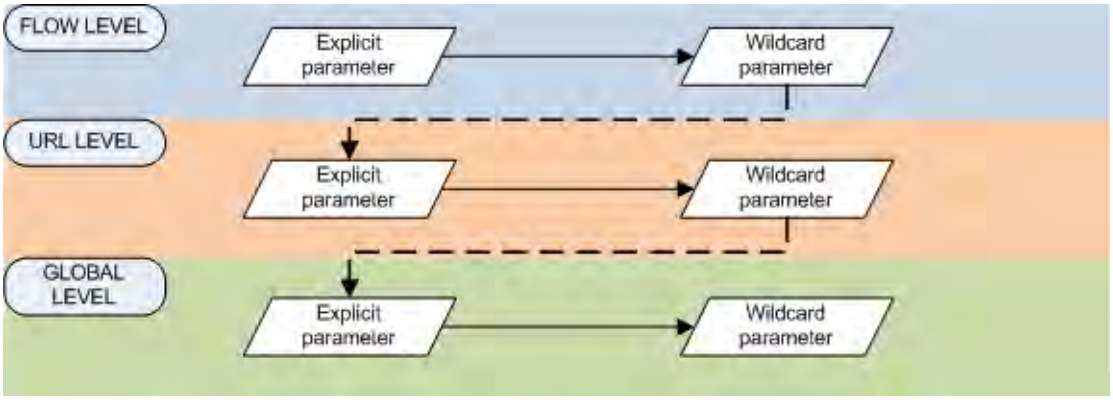


Figure 9.1 *Security Enforcer match process for parameters*

Tip

When adding wildcard URLs, you should arrange them in the order in which you want them to be enforced. The system enforces them from the top down.

To set the enforcement order for wildcard parameters

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. On the menu bar, click **Wildcards Order**.
   The Wildcards Order screen opens.
3. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
4. In the Wildcards Order area, for the **Global Parameters Wildcards List**, the **URL Parameters Wildcards List**, and the **Flow Parameters Wildcards List** options, adjust the order of the wildcards in the list by using the **Up** and **Down** buttons for each option.
5. Click the **Save** button to save any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Using wildcards for allowed modified cookie headers

You can use wildcards for allowed modified cookie headers to reduce the number of **Modified domain cookie** violations you receive when creating a security policy. For more information on cookie headers, refer to *Configuring allowed modified cookies*, on page 6-36.

You can enable tightening on wildcard cookies to cause the system to build the security policy by adding explicit cookies. When all of the explicit cookies have been learned, you (or the Policy Builder, if using automatic policy building) can delete the wildcard entity.

To use a wildcard for an allowed modified cookie

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. Above the Allowed Modified Cookies area, click the **Create** button.
   The New Allowed Cookie screen opens.
4. From the **Cookie Name Type** list, select **Wildcard**.
5. In the **Cookie Name** box, type the pattern string that matches the cookie name.

   ***Tip**: For wildcard syntax, refer to **Understanding wildcard syntax**, on page 9-1.*
6. If you want the system to add explicit cookies that match the wildcard cookie, check the **Tightening** box.
7. Click the **Create** button.
   The screen refreshes, and you can see the newly created allowed cookie in the Allowed Modified Cookies area.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area, then click **OK** to confirm.
   The system applies the updated security policy.

Checking the status of wildcard tightening for allowed modified cookies

You can check the tightening status of the cookies on the list of allowed modified cookies. If the wildcard allowed modified cookie has been tightened, the system displays a light bulb icon. The color of the light bulb provides details about the status of the allowed modified cookie:

- **Green**
  Indicates that no learning suggestions are available, but the tightening period is not over.
- **Yellow**
  Indicates that learning suggestions are available. Move the cursor over the light bulb icon to view whether the tightening period is over, or not.
- **Orange**
  Indicates that no learning suggestions are available and the tightening period is over. This entity is ready to be taken out of tightening, and be enforced.

We recommend you take an entity out of tightening when its tightening period is over, and you validate that requests for this entity did not log any suggestions.

To check the status of wildcard cookies

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those you are interested in.
3. If you want to search for cookies containing a specific string, after for the **Cookie Name Contains** setting, type the string.
4. For **Cookie Name Type**, select **Wildcard**.
5. For **Tightening**, specify the status of the cookies you want to display:
   - To view the cookies that are enabled in the security policy, select **Enabled**.
   - To view the cookies that are ready to be enforced in the security policy, select **Ready to be enforced**.
   - To view all of the cookies, select **All**.
6. Click **Go** to cause the system to filter the list of allowed modified cookies.
   The system updates the list of allowed modified cookies.

7. In the Tightening column of the allowed modified cookies list, point to the light bulb icon.
   The system displays information on the last time you or the Policy Builder tightened this wildcard entity (the last tightening event time).
8. If the status indicates that learning suggestions are available for any of the allowed modified cookies, in the navigation pane, point to **Manual Policy Building**, then click **Staging-Tightening Summary**.
   The Staging-Tightening Summary screen opens.
9. In the Allowed Cookies row, click the number in the Have Suggestions column.
   Learning suggestions for that cookie are displayed.
10. Review the suggestions that match the wildcard, decide which are legitimate for the web application, and accept them to the security policy.

Enforcing allowed modified cookie wildcards

You can delete wildcards for allowed modified cookies that you do not need in the security policy by enforcing the cookies. If a cookie in the allowed modified cookies list has an orange light bulb status, the tightening period is over and no more learning suggestions are available. In this case, you may want to enforce the allowed modified cookie.

To enforce allowed modified cookie wildcards

1. In the navigation pane, expand **Application Security** and click **Headers**.
   The Cookies screen opens.
2. In the editing context area, ensure that the edited web application and security policy are those that you want to update.
3. In the Allowed Modified Cookies area, in the Select column (far left), check the box next to the wildcard cookie that has tightening enabled and which you want to remove, click the **Enforce** button, and then confirm.
   The system deletes the wildcard cookie from the configuration.
4. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

10

Working with Parameters

- Understanding parameters
- Working with global parameters
- Working with URL parameters
- Working with flow parameters
- Configuring parameter characteristics
- Working with dynamic parameters and extractions
- Working with the parameter character sets
- Configuring sensitive parameters
- Configuring navigation parameters

Understanding parameters

Parameters are an integral entity in any web application. When you define wildcard or explicit parameters in a security policy, you are increasing the security of the web application. Application Security Manager™ evaluates defined parameters, meta characters, query string lengths, and POST data lengths as part of a positive security logic check. The Security Enforcer verifies the parameters that you configure in a security policy.

You can define parameters as global parameters, URL parameters, and flow parameters. For information on configuring global parameters, see *Working with global parameters*, on page 10-2. For information on configuring URL parameters, see *Working with URL parameters*, on page 10-5. For information on configuring flow parameters, see *Working with flow parameters*, on page 10-8.

You can create parameters containing different value types: static content, dynamic content, dynamic name, user-input, or XML value. You can also create parameters for which the system does not check or verify the value. You can configure a global, URL, or flow parameter as any value type with the exception of dynamic parameter names. With the exception of dynamic parameter names, y. The dynamic parameter name type is available only for flow parameters. Refer to *Understanding parameter value types*, on page 10-12, for more information.

When you create any type of parameter, the system automatically places the parameter in staging and does not block requests even if a violation occurs and the system is configured to block that violation. The system makes learning suggestions that you can accept or clear (see Chapter 13, *Refining the Security Policy Using Learning*). If you create wildcard parameters, you also have the option of enabling tightening.

This chapter discusses configuring explicit parameters. In Application Security Manager, you can also use wildcards for parameters. Refer to *Configuring wildcard parameters*, on page 9-13, for more information.

Understanding how the Security Enforcer processes parameters

The Security Enforcer uses the following order of priority when enforcing parameters:

- Flow parameters
- URL parameters
- Global parameters

If a parameter is defined more than once in the request context, the Security Enforcer applies only the more specific definition. For example, the parameter **param_1** is defined as a static content global parameter, and also defined as a user-input URL parameter. When the Application Security Manager receives a request for the parameter in a URL and the parameter is defined on both the global and URL level, the Security Enforcer generates any violations based on the URL parameter definition.

Working with global parameters

Global parameters are those that do not have an association with a specific URL or application flow. The advantage of using global parameters is that you can configure a global parameter once, and the Security Enforcer enforces the parameter wherever it occurs.

When you first create a global parameter, the system automatically places the parameter in staging and does not block requests even if a violation occurs and the system is configured to block violation. The system makes learning suggestions that you can accept or clear (see Chapter 13, *Refining the Security Policy Using Learning*). If you create wildcard global parameters, you also have the option of enabling tightening.

Creating a global parameter

You create a global parameter to address these conditions:

- The web application has a parameter that appears in several URLs or flows.
- You want the Application Security Manager to enforce the same parameter attributes across all parameters.

To create a global parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Above the Parameters List area, click the **Create** button.
   The Add Parameter screen opens.
4. In the Create New Parameter area, for the **Parameter Name** setting, select an option:
   - If you select **Explicit**, then in the box, type a unique parameter name.
   - If you select **Wildcard**, then in the box, type a pattern string that represents the parameter names. See *Configuring wildcard parameters*, on page 9-13, for more information.
   - If you select **No Name**, the system creates a parameter with the label, **UNNAMED**.
5. For the **Parameter Level** setting, select **Global Parameter**.
6. If you want the parameter to be in staging, leave the **Perform Staging** box checked.

7. If you are creating a wildcard parameter and you want the system to display explicit parameters that match the wildcard entity pattern that you specify, clear the **Perform Staging** box, and then check the **Perform Tightening** box.

   ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*

8. Specify whether the parameter requires a value:
   - If the parameter is acceptable without a value, leave the **Allow Empty Value** setting checked. (See *Creating parameters without defined values*, on page 10-20, for details.)
   - If the parameter must include a value, clear the check box.
9. To allow users to send a request that contains multiple parameters with the same name, check the **Allow Repeated Occurrences** box. The default setting is disabled.
10. If you want to treat the parameter you are creating as a sensitive parameter (not visible in logs or the user interface), check **Sensitive Parameter**.
11. For the **Parameter Value Type** setting, select the format for the parameter value. Depending on the value type you select, the screen refreshes to display additional configuration options. See *Understanding parameter value types*, on page 10-12, for information on parameter types and additional settings that are associated with them.
12. Click the **Create** button to add the new global parameter to the security policy.
    The screen refreshes, and displays the new global parameter.
13. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
    The system applies the updated security policy.

Editing the properties of a global parameter

At times, you may want to update the characteristics of a global parameter. This is easily done by editing the parameter properties.

Note

You cannot change the name of a parameter.

To edit a global parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, click the name of the parameter whose properties you want to edit.
   The Parameter Properties screen opens.
4. Make any changes to the parameter properties, as required.
5. When you have finished, click the **Update** button.
   The system saves any changes you may have made, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Deleting a global parameter

Web applications can change over time, and you may want to delete a global parameter.

To delete a global parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, check the box next to the global parameter that you want to remove, and then click the **Delete** button.
   The system displays a popup confirmation screen.
4. Click **OK**.
   The system deletes the parameter.

Working with URL parameters

You define parameters in the context of a URL when a parameter is relevant to that particular URL, and you do not want the system to also verify the URL's associated flows. That is, you can use a URL parameter when it does not matter where users were before they access this URL or whether the parameter was in a GET or POST request.

Defining a parameter as a URL parameter allows you to control one or all of the parameters associated with that URL, and allows users to create exceptions, if needed, to wildcard or other global definitions. When you define a URL parameter, the Security Enforcer applies the security policy to the parameter attributes in the context of the associated URL, and ignores the flow information.

Note that when you first create a URL parameter, the system places the parameter in staging by default and does not block requests even if a violation occurs and the system is configured to block the violation. The system makes learning suggestions that you can accept or clear (see Chapter 13, *Refining the Security Policy Using Learning*). If you create wildcard URL parameters, you also have the option of enabling tightening.

Creating a URL parameter

When you create a parameter that is associated with a URL, the Security Enforcer verifies the parameter in the context of the URL.

◆ Note

The prerequisite for this task is that the security policy already includes the URL for which you want to add a parameter. If the security policy does not yet include the URL, refer to ***Configuring URLs****, on page 6-21, for information on adding a URL to the configuration.*

To create a parameter associated with a URL

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Above the Parameters List area, click the **Create** button.
   The Add Parameter screen opens.

4. In the Create New Parameter area, for the **Parameter Name** setting, select an option:
   - If you select **Explicit**, then in the box, type a unique parameter name.
   - If you select **Wildcard**, then in the box, type a pattern string that represents the parameter names. See *Configuring wildcard parameters*, on page 9-13, for more information.
   - If you select **No Name**, the system creates a parameter with the label, **UNNAMED**.
5. For the **Parameter Level** setting, select **URL Parameter**.
   The screen refreshes and displays the **URL Path** option.
   - For the **URL Path** option, select a protocol from the list, and then type the URL in this format:

     `/url_name.ext`
6. If you want the parameter to be in staging, leave the **Perform Staging** box checked.
7. If you are creating a wildcard parameter and you want the system to display explicit parameters that match the wildcard entity pattern that you specify, clear the **Perform Staging** box, and then check the **Perform Tightening** box.

   ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*
8. Specify whether the parameter requires a value:
   - If the parameter is acceptable without a value, leave the **Allow Empty Value** setting checked. (See *Creating parameters without defined values*, on page 10-20, for details.)
   - If the parameter must include a value, clear the check box.
9. To allow users to send a request that contains multiple parameters with the same name, check the **Allow Repeated Occurrences** box. The default setting is disabled.
10. If you want to treat the parameter you are creating as a sensitive parameter (not visible in logs or the user interface), check **Sensitive Parameter**.
11. For the **Parameter Value Type** setting, select the format for the parameter value.
    Depending on the value type you select, the screen refreshes to display additional configuration options. See *Understanding parameter value types*, on page 10-12, for information on parameter types and additional settings that are associated with them.
12. Click the **Create** button to add the new URL parameter to the security policy.
    The screen refreshes, and displays the new URL parameter.

13. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
The system applies the updated security policy.

Editing the properties of a URL parameter

At times, you may want to update a URL parameter. This is easily done by editing the parameter properties.

Note

You cannot change the name of a parameter.

To edit the properties of a URL parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the parameter whose properties you want to edit.
The Parameter Properties screen opens.
4. Make any changes to the parameter properties, as required.
5. When you have finished, click the **Update** button.
The system saves any changes you may have made, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
The system applies the updated security policy.

Deleting a URL parameter

Web applications can change over time, and there may be occasions when you want to delete a parameter from the security policy.

To delete a parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, check the box next to the parameter that you want to remove, and then click the **Delete** button.
The system displays a popup confirmation screen.

4. Click **OK**.
   The system deletes the parameter.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Working with flow parameters

You define parameters in the context of a flow when it is important to enforce that a target URL receives a parameter from a specific referrer URL. Defining a parameter in the context of a flow is the most specific context, and thus provides the tightest definition for the web application.

When you first create a flow parameter, the system automatically places the parameter in staging and does not block requests even if a violation occurs and the system is configured to block the violation. The system makes learning suggestions that you can accept or clear (see Chapter 13, *Refining the Security Policy Using Learning*). If you create wildcard flow parameters, you also have the option of enabling tightening.

Creating a flow parameter

When you create a parameter that is associated with a flow, the Security Enforcer verifies the parameter in the context of the flow (see *Configuring flows*, on page 6-30, for more information). For example, if you define a parameter in the context of a GET request, and a client sends a POST request that contains the parameter, the Security Enforcer generates an **Illegal Parameter** violation.

You can define flow parameters for very tight, flow-specific security. With this increased protection comes an increase in maintenance and configuration time. Note that if your web application uses dynamic parameters, you manually add those to the security policy.

The following task starts after the flow for which you want to create a parameter is configured. If the security policy does not include the flow, refer to *Configuring flows*, on page 6-30, for information on adding a flow to the configuration.

To create a parameter associated with an application flow

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Above the Parameters List area, click the **Create** button.
   The Add Parameter screen opens.

4. In the Create New Parameter area, for the **Parameter Name** setting, select an option:
   - If you select **Explicit**, then in the box, type a unique parameter name.
   - If you select **Wildcard**, then in the box, type a pattern string that represents the parameter names. See *Configuring wildcard parameters*, on page 9-13, for more information.
   - If you select **No Name**, the system creates a parameter with the label, **UNNAMED**.
5. For the **Parameter Level** setting, select **Flow Parameter**.
   The screen refreshes and displays flow detail settings.
6. For the **From URL** setting:
   - If the source URL is an entry point, click **Entry Point**.
   - If the source URL is a referrer URL (the referrer URL must already be defined in the policy), click **URL Path**, select the protocol used to request the URL, then type the referrer URL associated with the flow.
7. For the **Method** setting, select the HTTP method that applies to the target URL (the referrer URL must already be defined in the policy).
8. For the **To URL** setting, if you specified a referrer URL for the **From URL** setting, specify the target URL.
9. If you want the parameter to be in staging, leave the **Perform Staging** box checked.
10. If you are creating a wildcard parameter and you want the system to display explicit parameters that match the wildcard entity pattern that you specify, clear the **Perform Staging** box, and then check the **Perform Tightening** box.

    ***Note**: F5 Networks recommends against using both tightening and staging at the same time on the same wildcard entity.*
11. If the parameter is required in the context of the flow, check the **Is Mandatory Parameter** setting. Note that only flows can have mandatory parameters. (See *Allowing multiple occurrences of a parameter in a request*, on page 10-21, for more information.)
12. Specify whether the parameter requires a value:
    - If the parameter is acceptable without a value, leave the **Allow Empty Value** setting checked. (See *Creating parameters without defined values*, on page 10-20, for details.)
    - If the parameter must include a value, clear the check box.
13. To allow users to send a request that contains multiple parameters with the same name, check the **Allow Repeated Occurrences** box. The default setting is disabled.
14. If you want to treat the parameter you are creating as a sensitive parameter (not visible in logs or the user interface), check **Sensitive Parameter**.

15. For the **Parameter Value Type** setting, select the format for the parameter value. Depending on the value type you select, the screen refreshes to display additional configuration options. See *Understanding parameter value types*, on page 10-12, for information on parameter types and additional settings that are associated with them.
16. Click the **Create** button to add the new flow parameter to the security policy.
    The screen refreshes, and displays the new flow parameter.
17. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
    The system applies the updated security policy.

Editing the properties of a flow parameter

At times, you may want to update the characteristics of a flow parameter. This is easily done by editing the parameter properties.

Note

You cannot change the name of a parameter.

To edit the properties of a flow parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the parameter whose properties you want to edit.
   The Parameter Properties screen opens.
4. Make any changes to the parameter properties, as required.
5. When you have finished, click the **Update** button.
   The system saves any changes you may have made, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Deleting a flow parameter

Web applications can change over time, and there may be occasions when you want to delete a parameter from a flow.

To delete a parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Select column (far left), check the box next to the parameter that you want to remove, and then click the **Delete** button.
   The system displays a popup confirmation screen.
4. Click **OK**.
   The system deletes the parameter.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring parameter characteristics

Parameter characteristics define the individual attributes of the parameter. The parameter characteristics change depending on the type of parameter that you specify.

Understanding parameter value types

When you add a parameter to the security policy, you specify the parameter value type. The Security Enforcer then knows in what form to expect the parameter value, and applies the security policy accordingly.

You can configure global parameters, URL parameters, and flow parameters as any parameter type, except the dynamic parameter name type. You can configure only flow parameters as dynamic parameter names.

The parameter value types are:

- **Ignore value**
  If you do not want the system to perform checks on the parameter value, use this parameter value type.
- **Static content value**
  Static parameters are those that have a known set of values. A list of country names or a yes/no form field are both examples of static parameters. If you select this type, you add or remove static values for the parameter. For information on configuring static parameters, see *Configuring static parameters*, on page 10-13.
- **Dynamic content value**
  Dynamic parameters are those whose set of values can change, and are often linked to a user session. When you create a new parameter of this type, you are prompted to define dynamic parameter extraction properties. The server sets the value for dynamic content value (DCV) parameters. DCV parameters are often associated with applications that use session IDs for client sessions. For information on configuring DCV parameters, see *Configuring dynamic content value parameters*, on page 10-24.
- **Dynamic parameter name**
  Some flow parameters have names that change dynamically. If so, you can use this parameter type. If you select this type, you also need to specify the URL from which the system should extract dynamic parameter name parameters. For information on configuring dynamic parameter names, see *Configuring parameter characteristics for dynamic parameter names*, on page 10-27.
- **User-input value**
  User-input parameters are those that require users to enter or provide some sort of data. This is the most commonly used parameter value type. Comment, name, and phone number fields on an online form are all examples of user-input parameters. You can also configure user-input parameters even if the parameter is not really user input. For example, if a parameter has a wide range of values or many static values, you may

want to configure the parameter as a user-input parameter instead of a static content parameter. For information on configuring user-input parameters, see *Configuring parameter characteristics for user-input parameters*, on page 10-14.

- **XML value**
  XML parameters are those whose parameter value contains XML data. For information on configuring XML parameters, see *Associating an XML profile with a parameter*, on page 12-22.

Configuring static parameters

Static parameters are parameters that can contain values from a specific set. For example, a credit card type parameter, for payment in a shopping application, may have the value set of **MasterCard**®, **Visa**®, and **American Express**®. When you configure static parameters, you are basically creating a value set for the parameter.

To configure static parameters

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, select **Static content value**. The screen refreshes and displays the Parameter Static Values area.
3. In the Parameter Static Values area, in the **New Static Value** box, type a value for the parameter.
4. Click the **Add** button to add the value to the Parameter Static Values list.
5. Repeat steps 3 and 4 to add all the values that this parameter can have.
6. Click the **Create** button to save the parameter in the configuration.
7. In the editing context area, click the **Apply Policy** button to immediately put the security policy changes into effect.

Configuring parameter characteristics for user-input parameters

User-input parameters are those for which the user can provide a value. For user-input parameters, you can configure the Application Security Manager to verify minimum and maximum values, minimum and maximum lengths, and valid meta characters. It is particularly useful to configure a parameter as a user-input parameter if you want the system to verify parameter values using broad validations, such as minimum and maximum value or maximum length.

By default, the system checks for attack patterns within all user-input alpha-numeric parameters. For each parameter, you can enable or disable a specific attack signature.

User-input parameters can accept many different data types. The data types are: alpha-numeric, binary, decimal, email, integer, and phone. Depending on the data type that you configure, the system can verify additional options, as noted in the following sections.

◆ Tip

A valuable characteristic of user-input parameters is the ability to attach attack signatures to them.

Configuring an alpha-numeric user-input parameter

The alpha-numeric data type specifies that the parameter value can have letters, integers, and the underscore character in it. For this data type, you can specify a maximum length, and you can define the acceptable parameter values as a regular expression. You can also specify one or more meta characters (in addition to the base character set of **a-z**, **A-Z**, **0-9)**, and one or more regular expressions, that are acceptable within the context of the parameter.

◆ Note

If you enable regular expressions for an alpha-numeric parameter, it results in a mismatch that generates a ***Parameter value does not comply with regular expression*** *violation.*

To configure an alpha-numeric user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.

3. For the **Data Type** setting, use the default value, **Alpha-Numeric**.
   - To enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
   - To enforce the parameter value using pattern matching, check the **Regular Expression** box, and type a regular expression.

     Note: *When you enable this setting, the only values acceptable for the parameter are those that exactly match the regular expression pattern that you provide. All other values are considered illegal for this parameter.*
4. If you want to make certain meta characters valid, or not valid, as part of the parameter value (and override the global meta character settings), click **Value Meta Characters**.
   - Make sure that the **Check characters on this parameter** check box is checked.
     The screen displays the global and overridden meta character settings for this parameter.
   - From the **Global Security Policy Settings** list, select any meta characters that you want to assign to the parameter value, and click the Move button (<<) to add them to the **Overridden Security Policy Settings** list.
     The screen displays the meta characters and the default state for each.
   - In the **Overridden Security Policy Settings list**, change the meta character state as required.
     - Select **Allowed** when the meta character can be in the parameter value.
     - Select **Disallowed** when the meta character cannot be in the parameter value, and may trigger the **Illegal meta character in parameter value** violation.
5. If you want to make certain known attack patterns valid, or not valid, as part of the parameter value, click **Attack Signatures**.
   - Make sure that the **Check attack signatures on this parameter** check box is checked.
     The screen displays the attack signature settings that are available or assigned to this parameter.
   - From the Global Security Policy Settings list, select any attack signatures that you want to assign to the parameter value, and click the Move button (<<) to add them to the Overridden Security Policy Settings list.
     The screen displays the attack signatures and the default state for each.

- In the Overridden Security Policy Settings list, change the attack signature state as required. Note that the state that you select may override the state that is assigned at the attack signature set level.
  - Select **Disabled** when the parameter value can match the attack signature.
  - Select **Enabled** when the parameter value cannot match the attack signature.

6. Click the **Create** button to add the parameter to the configuration.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring a binary user-input parameter

The binary data type specifies that the parameter value is data for which the system does not verify meta characters or attack. Typically, you use this data type for binary file uploads. Note that for this data type, you specify only a maximum length.

To configure a binary user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.
3. For the **Data Type** setting, select **Binary (Length checks only)**.
4. If you want the Security Enforcer to enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
5. Click the **Create** button to add the parameter to the configuration.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring a decimal user-input parameter

The decimal data type specifies that the parameter value is numeric, and can include integers and decimals only. For this data type, you can specify a minimum value, a maximum value, and a maximum length.

To configure a decimal user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.
3. For the **Data Type** setting, select **Decimal**.
4. If you want to enforce a minimum value for the parameter, check the **Check Minimum Value** box, and type a number.
5. If you want to enforce a maximum value for the parameter value, check the **Check Maximum Value** box, and type a number.
6. If you want to enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
7. Click the **Create** button to add the parameter to the configuration.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring an email user-input parameter

The email data type specifies that the parameter value is in the email address format. Values for this data type can include letters, numbers, the at meta character (**@**), the period (**.**) character, and the underscore (**_**) character. For this data type you can specify only a maximum length.

◆ Note

F5 Networks recommends that you use the email data type only if the web application has client-side data validation for the parameter.

To configure an email user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.
3. For the **Data Type** setting, select **Email**.
4. If you want the Security Enforcer to enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
5. Click the **Create** button to add the parameter to the configuration.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring an integer user-input parameter

The integer data type specifies that the parameter value is numeric, and can include only whole numbers. For this data type, you can specify a minimum value, a maximum value, and a maximum length.

To configure an integer user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.
3. For the **Data Type** setting, select **Integer**.
4. If you want the Security Enforcer to enforce a minimum value for the parameter value, check the **Check Minimum Value** box, and type a number.
5. If you want the Security Enforcer to enforce a maximum value for the parameter value, check the **Check Maximum Value** box, and type a number.

6. If you want the Security Enforcer to enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
7. Click the **Create** button to add the parameter to the configuration.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring a phone user-input parameter

The phone data type specifies that the parameter value is in the phone number format. Values for this data type can include numbers, the hyphen meta character (-), and the parentheses meta characters [**( )**]. For this data type you can specify only a maximum length.

◆ Note

F5 Networks recommends that you use the phone data type only if the web application has client-side data validation for the parameter.

To configure a phone user-input parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, use the default value, **User-input value**.
3. For the **Data Type** setting, select **Phone**.
4. If you want to enforce a maximum length (number of bytes) for the parameter value, check the **Check Maximum Length** box, and type a number.
5. Click the **Create** button to add the parameter to the configuration.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Creating parameters without defined values

The **Allow Empty Value** setting specifies whether the system expects the parameter to have a defined value. When this setting is enabled on a parameter (which is the default setting), the system does not generate an **Illegal empty parameter value** alert if a client request does not provide a value. Conversely, if the **Allow Empty Value** setting is disabled, the system generates the **Illegal empty parameter value** alert if a client request does not provide a value. The **Allow Empty Value** setting is applicable to global parameters, URL parameters, and flow parameters.

To set the Allow Empty Value setting

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the parameter whose properties you want to edit.
   The Parameter Properties screen opens.
4. For the **Allow Empty Value** setting, check or clear the check box as required.
5. When you have finished, click the **Update** button.
   The system saves any changes you may have made, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Allowing multiple occurrences of a parameter in a request

By sending several occurrences of the same parameter in a single request, an attacker can cause unexpected behavior on an application server. This type of attack, called ***HTTP parameter pollution***, can be used for web application firewall evasion (and can allow smuggling attacks through intrusion prevention signature matching engines).

Since most web applications do not expect parameters to appear several times in requests, such behavior is not allowed, by default. Therefore, when a request contains multiple occurrences of the same parameter, the system generates an **Illegal repeated parameter name** violation (if that violation is set to Alarm or Block). If the violation occurs, the system provides a learning suggestion that you can review to decide whether to allow repeated occurrences of the parameter. You can also enable the **Allow Repeated Occurrences** setting by editing parameter properties.

To allow repeated occurrences of a parameter in a request

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the parameter that you want to edit.
   The Parameter Properties screen opens.
4. Check the **Allow Repeated Occurrences** box.
5. Click the **Update** button.
   The system saves the changes, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Making a flow parameter mandatory

The **Is Mandatory Parameter** setting specifies whether a parameter must be present in a flow.

Note

You can configure only flow parameters as mandatory.

To make a flow parameter mandatory

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, in the Parameter Name column, click the name of the flow parameter whose properties you want to edit.
   The Parameter Properties screen opens.
4. Check the **Is Mandatory Parameter** check box.
5. When you have finished, click the **Update** button.
   The system saves any changes you may have made, and returns you to the Parameters List screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring XML parameters

XML parameters contain XML data in the parameter value. To perform checks on the XML data, you associate an XML profile with the XML parameter. For details on configuring XML profiles, refer to Chapter 12, *Protecting XML Applications*.

To create an XML parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, select **XML value.**
   The screen refreshes and displays additional settings.
3. For the **XML Profile** setting, perform the appropriate task:
   - If you already created an XML profile, select it from the list.
   - If you have not created an XML profile, click the Create button (+) next to **XML Profile** to create one. For details about creating XML profiles, refer to Chapter 12, *Protecting XML Applications*.
4. Click the **Create** button.
   The screen refreshes and you see the parameter in the list.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Working with dynamic parameters and extractions

When you configure a dynamic parameter, you also configure the extraction properties for the parameter values. The extraction properties define from where to extract the dynamic parameter values or name, and which method or methods to use for the extraction. When the Application Security Manager receives a request that contains an entity (for example, a file extension or URL) containing a dynamic parameter, the system uses the extraction properties to collect the parameter value or name from web application's response to the request. Once the system has extracted the dynamic parameter values, the Security Enforcer knows what to enforce the next time a request contains the dynamic parameter.

Configuring dynamic content value parameters

Dynamic content value (DCV) parameters are those for which the web application sets the value on the server side. When you configure a DCV parameter in the Application Security Manager, the system verifies that the client is not changing the parameter value, as set by the server, from one request to the next. For example, in an auction application, you might configure the price parameter as a DCV parameter to keep users from tampering with the price.

DCV parameters are often associated with web applications that use sessions. Each user of these applications has unique identifiers, and those identifiers may also change. As a result, the parameters in the web application that identify the user have dynamic content values. As an example, user identity is often passed between pages as a "hidden" parameter, which could be exploited by malicious users.

When you configure a DCV parameter, you also configure the extraction properties for the parameter values. The extraction properties specify the manner in which the Application Security Manager discovers and populates the values for the DCV parameter.

By default, the system retains all of the values that it finds for a DCV parameter unless the number of values exceeds 950. When that is the case, the Application Security Manager replaces the first-extracted values with new values. When there are fewer than 950 values, the system does not replace the values it knows about when it extracts a new value.

To configure a dynamic content value parameter

1. Create a new parameter.
   - To create a global parameter, see *Creating a global parameter*, on page 10-2.
   - To create a URL parameter, see *Creating a URL parameter*, on page 10-5.
   - To create a flow parameter, see *Creating a flow parameter*, on page 10-8.
2. For the **Parameter Value Type** setting, select **Dynamic content value**.
3. Click the **Create** button.
   A popup screen opens asking if you want to define extractions.
4. Click **OK**.
   The Create New Extraction screen opens.
5. For the **Name** setting, select a name for the dynamic parameter or type a name.
6. From the **Extracted Items Configuration** list, accept **Basic** (default) or select **Advanced**, and then specify from where you want the system to extract the dynamic parameter values.
   For more information on this setting, see *Understanding the extracted items configuration*, on page 10-26.
7. From the **Extraction Methods Configuration** list, select **Basic** or **Advanced**, and then specify the method or methods that you want the system to use to extract the dynamic parameter values.
   For more information on this setting, see *Understanding the extraction methods configuration*, on page 10-26.
8. Click the **Create** button to add the extraction properties to the parameter.
9. Click the **Update** button to update the parameter settings.
10. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
    The system applies the updated security policy.

◆ Note

You should define the extractions for a DCV parameter before you apply the security policy that includes the parameters. If you do not, when you apply the security policy, the policy validator generates a warning that the security policy contains dynamic parameters that do not have extractions defined.

Understanding the extracted items configuration

When you create an extraction for a dynamic parameter, one aspect of the extraction is configuring where, in the responses of request objects, the system searches for the dynamic parameter. You can configure the system to extract the dynamic parameter values from file types, URLs, and by using pattern matching. Alternately, you can configure the system to extract dynamic parameter values from all items. Table 10.1 describes the extracted items settings.

Extraction item	Description
File Types	Use this setting when you want the system to extract dynamic parameters from files of a certain type. Note that the available file types are those that are already a part of the security policy.
URLs	Use this setting when you want the system to extract dynamic parameters from specific URLs.
RegExp	Use this setting when you want the system to extract dynamic parameters that match a regular expression pattern. Note that this setting is available only when you select **Advanced** (above the Extracted Items Configuration area).
Extract From All items	Use this setting when you want the system to extract dynamic parameters from all text-based URLs and file types. Note that this setting is available only when you select **Advanced** (from the **Extracted Items Configuration** list).

Table 10.1 *Extraction locations for dynamic parameters*

Understanding the extraction methods configuration

Another important aspect of the extraction configuration is defining how the system extracts the dynamic parameter, that is, the extraction method. Table 10.2 describes the extraction methods.

Extraction method	Description
Search in Links	Use this setting when you want the system to extract dynamic parameter values from links (**href** tags) within the server response to a URL.
Search Entire Form	Use this setting when you want the system to extract dynamic parameter values from all parameters in all forms in the HTML response to a requested URL.
Search Within Form	Use this setting when you want the system to extract dynamic parameter values from a specific parameter within in a form. Note that this setting is available only when you select **Advanced** (from the **Extracted Items Configuration** list).

Table 10.2 *Extraction methods for dynamic parameters*

Extraction method	Description
Search in XML	Use this setting when you want the system to extract dynamic parameter values from within XML entities. Note that this setting is available only when you select **Advanced** (from the **Extraction Methods Configuration** list).
Search in Response Body	Use this setting when you want to specify where in the response the system is to search dynamic parameter values for extraction. Note that this setting is available only when you select **Advanced** (from the **Extraction Methods Configuration** list).

Table 10.2 *Extraction methods for dynamic parameters (Continued)*

Viewing the list of extractions

You can review all of the parameter extractions that are configured in the security policy. You can also review the parameter extractions for a specific URL on the properties screen for that URL. See *Configuring URLs*, on page 6-21, for more information on URL properties.

To view the configured extractions

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. On the menu bar, click **Extractions**.
   The Extractions screen opens, where you can view the extractions that are in the security policy.

Configuring parameter characteristics for dynamic parameter names

In some web applications, DCV parameters also have dynamic names. You can use the parameter type, **Dynamic parameter name**, when you want the Security Enforcer to apply the dynamic names as well as dynamic values. Note that the **Dynamic parameter name** parameter type is applicable only when you are configuring a flow parameter.

When you configure a dynamic parameter name, you also configure the extraction properties. The extraction properties specify the manner in which the Application Security Manager discovers the parameter names.

To configure a dynamic parameter name parameter

1. Create a flow parameter. (See *Creating a flow parameter*, on page 10-8).
2. In the Create New Parameter area, for the **Parameter Value Type** setting, select **Dynamic parameter name**.
   The screen refreshes, automatically generates a unique name in the **Parameter Name** setting, and displays the Dynamic Parameter Properties area.

3. In the Dynamic Parameter Properties area, for the **Extract Parameter from URL** setting, select the protocol to use and type the URL from which you want the system to extract the dynamic parameter.
4. Next, select whether the system searches for the parameter in a form, or in the response body.
   - If the parameter is located in a form, select **Search Within Form**, and specify the form index and parameter index.
   - If the parameter is located in the HTTP/S response, select **Search parameters in response body (in form elements names only)**.
     - In the **By Pattern** box, type a regular expression that represents the parameter name pattern.
     - If you do not want the system to enforce whether the parameter has a value, clear the **Check parameter value** box.
5. Click the **Create** button to add the new parameter to the configuration.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Working with the parameter character sets

Each security policy includes a default character set for parameter names and another for parameter values. The default character sets correspond to the language encoding that you specified for the web application. The Security Enforcer implements the character set based on the state of the character or meta character: **Allowed** or **Disallowed**.

You can change the enforcement state for the general character set, or within the context of a specific alpha-numeric user-input parameter. For alpha-numeric user-input parameters, you can also specify which characters or meta characters are enforced, as well as override the default state. For more information on configuring alpha-numeric user-input parameters, see *Configuring an alpha-numeric user-input parameter*, on page 10-14.

Viewing and modifying the default parameter value character set

The parameter value character set controls the default characters and meta characters that are acceptable in a parameter value.

To view or modify the default parameter value character set

1. In the navigation pane, expand **Application Security**, point to **Parameters,** point to **Character Sets**, and then click **Parameter Value**.
   The Parameter Value Character Set screen opens showing the default character set.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Use the **Filter** option to display the characters or meta characters that you want to view.
4. For each character or meta character, change the state, as required.
   - **Allowed**: Specifies that the security policy permits this character or meta character in parameter values.
   - **Disallowed**: Specifies that the security policy does not permit this character or meta character in parameter values.
5. Click the **Save** button to save any changes you may have made on this screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Viewing and modifying the default parameter name character set

The parameter name character set controls the default characters and meta characters that are acceptable in a parameter name.

To view or modify the default parameter name character set

1. In the navigation pane, expand **Application Security**, point to **Parameters,** point to **Character Sets**, and then click **Parameter Name**.
   The Parameter Name Character Set screen opens showing the default character set for wildcard parameter names.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Use the **Filter** option to display the characters or meta characters that you want to view.
4. For each character or meta character, change the state, as required.
   - **Allowed**: Specifies that the security policy permits this character or meta character in parameter values.
   - **Disallowed**: Specifies that the security policy does not permit this character or meta character in parameter values.
5. Click the **Save** button to save any changes you may have made on this screen.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

Configuring sensitive parameters

The Application Security Manager stores incoming requests in plain text format. Some requests include sensitive data in parameters, such as an account number. If you create sensitive parameters. the system replaces the sensitive data, in the stored request and in logs, with asterisks (***).

You can create sensitive parameters as described in the procedure, following, or by checking the **Sensitive Parameter** box when creating or editing any parameter. All parameters defined as sensitive, regardless of how you configured them, appear in the Sensitive Parameters list.

Configuring a parameter as sensitive affects only how the Application Security Manager stores and displays information in requests. It does not affect requests sent to the web application or the client.

◆ Note

*The Application Security Manager automatically creates a sensitive parameter called **password** for every new security policy.*

To create a sensitive parameter

1. In the navigation pane, expand **Application Security**, point to **Parameters,** then click **Sensitive Parameters**.
   The Sensitive Parameters screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. Above the Sensitive Parameters section, click the **Create** button.
   The New Sensitive Parameter screen opens.
4. In the **Parameter** box, type the name of the user-input parameter, exactly as it occurs in the HTTP request, for which you do not want the system to store the actual value. In the following example, **account** is the sensitive parameter:

   ```
   http://www.siterequest.com/bank.php?account=12345
   ```

   ***Tip:** If a parameter of this name already exists in the security policy, click it in the parameter list, and check its **Sensitive Parameter** box instead of creating a new sensitive parameter.*
5. Click the **Create** button.
   The screen closes, and you can see the newly created sensitive parameter in the Sensitive Parameters list.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

In addition to creating sensitive parameters, you can also edit or delete existing sensitive parameters. To edit an existing sensitive parameter, click the name, then update the parameter settings. To delete a parameter, check the select box and click the **Delete** button.

Configuring navigation parameters

If you want the security policy to differentiate between pages in the web application that are generated by requests with the same URL name but with different parameter and value pairs, and to build the appropriate flows, you must specify the exact names of the parameters that trigger the creation of the pages in the web application. These parameters are known as ***navigation parameters***.

To specify a navigation parameter

1. In the navigation pane, expand **Application Security**, point to **Parameters** then click **Navigation Parameters**.
   The Navigation Parameters screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. Above the Navigation Parameters area, click the **Create** button.
   The New Navigation Parameter screen opens.
4. Specify the URL to which the navigation parameter applies:
   - If the new navigation parameter applies to every page in the web application, select **Any URL**.
   - If the navigation parameter applies to only one page in the web application, select **URL Path**, and type the URL.
5. In the **Navigation Parameter** box, type the name of the parameter passed to the web server for dynamic page-building purposes.
6. Click the **Create** button.
   The screen closes, and on the Navigation Parameters screen, you can see the new navigation parameter.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button, then click **OK** to confirm.
   The system applies the updated security policy.

In addition to creating navigation parameters, you can also edit or delete existing navigation parameters, as required by changes in the web application. To delete an existing navigation parameter, check the box next to the parameter, and click the **Delete** button. To edit an existing navigation parameter, click the name then update the parameter properties.

11

Working with Attack Signatures

- Overview of attack signatures
- Types of attacks that attack signatures detect
- Managing the attack signatures pool
- Updating the system-supplied attack signatures
- Working with attack signature sets
- Modifying the blocking policy for an attack signature set
- Understanding attack signature staging
- Managing user-defined attack signatures

Overview of attack signatures

Attack signatures are rules or patterns that identify attacks or classes of attacks on a web application and its components. You can apply attack signatures to both requests and responses. Additionally, within the requests signatures pool, some signatures can apply to alpha-numeric user-input parameters.

The attack signatures feature has the following characteristics:

- The system has a global attack signatures pool.
- You configure and schedule updates to the system-supplied attack signatures pool.
- The system has several predefined attack signature sets.
- You can define attack signature sets.
- Each security policy has its own attack signature set assignments.
- You can create user-defined attack signatures.
- You can import and export user-defined attack signatures.

Understanding the global attack signatures pool

The global attack signatures pool contains all of the attack signatures that are part of the Application Security Manager™ configuration. This includes both system-supplied attack signatures, including XML signatures, and user-defined attack signatures.

Overview of system-supplied attack signatures

The Application Security Manager ships with an extensive database of attack signatures. These are known as ***system-supplied attack signatures***. You can disable system-supplied attack signatures, but you cannot delete them.You can also update system-supplied attack signatures to ensure that the system always has the most current protection against known attacks. For information on updating the attack signatures pool, refer to *Updating the system-supplied attack signatures*, on page 11-10.

Overview of user-defined attack signatures

User-defined attack signatures are those written by customers. User-defined attack signatures must follow the same syntax rules as the system-supplied attack signatures. For details on creating and managing user-defined attack signatures, see *Managing user-defined attack signatures*, on page 11-25.

Overview of attack signature sets

An ***attack signature set*** is a group of individual attack signatures. Rather than applying individual attack signatures to a security policy, you can apply one or more attack signature sets. The Application Security Manager ships with several system-supplied signature sets. By default, a generic attack signature set is assigned to new security policies. Additionally, you can create your own attack signature sets. For information on creating and managing attack signature sets, refer to *Working with attack signature sets*, on page 11-13.

Understanding how the system uses attack signatures

Attack signatures apply to requests, responses, and alpha-numeric user-input parameters. Request signatures apply to the entire request, or the elements of the request. Response signatures are similar to request signatures, and provide an additional level of security for attacks that may have avoided detection in the request. Parameter signatures, which are a subset of the request signatures, apply to the name and value pairs for the alpha-numeric user-input parameters that are defined in a security policy. These signatures attempt to identify classes of attacks, for example, SQL injection, command injection, cross-site scripting, and directory traversal. Refer to *Types of attacks that attack signatures detect*, on page 11-3, for specific information on the various attack types.

When the Application Security Manager receives a request, the system applies the attack signatures associated with the security policy to the request. If, in the request (or response), a pattern matches one or more of the attack signatures, the system generates the **Attack signature detected** violation. If the enforcement mode is blocking, the system also blocks the request and issues the Blocking Response Page to the client making the request.

Types of attacks that attack signatures detect

Table 11.1 describes common web application attacks that attack signatures can detect.

Attack type	Description
Abuse of functionality	Abuse of functionality is an attack technique that uses a web site's own features and functionality to consume, defraud, or circumvent the application's access control mechanisms.
Authentication/authorization attacks	Authentication attacks target a web site's method of validating the identity of a user, service or application. Authorization attacks target a web site's method of determining if a user, service, or application has the necessary permissions to perform a requested action.
Brute force attack	A brute force attack is an outside attempt by hackers to access post-logon pages of a web site by guessing user names and passwords; brute force attacks are performed when a malicious user attempts to log on to a URL numerous times, running many combinations of user names and passwords until they successfully log on.
Buffer overflow	Buffer overflow exploits are attacks that alter the flow on an application by overwriting parts of memory. An attacker could trigger a buffer overflow by sending a large amount of unexpected data to a vulnerable component of the web server.
Command execution	Command execution attacks are those where an attacker manipulates the data for a user-input field, by submitting commands that could alter the web page content or web application by running a shell command on a remote server to reveal sensitive data—for example, a list of users on a server.
Cross-site scripting (XSS)	Cross-site scripting (XSS) is an attack technique that forces a web site to echo attacker-supplied executable code, which loads in a user's browser.
Denial of Service	Denial of Service (DoS) is an attack technique that overwhelms system resources to prevent a web site from serving normal user activity.
Detection evasion	Detection evasion is an attack technique that attempts to disguise or hide an attack to avoid detection by an attack signature.
Directory indexing	Automatic directory listing/indexing is a web server function that lists all of the files within a requested directory if the normal base file is not present.
Forceful browsing	Forced browsing is an attack where the aim is to list and access resources that the application does not directly reference, but are still accessible. An attacker can search for unlinked contents, such as temporary directories and files, and old backup and configuration files. These resources may contain sensitive information.
HTTP parser attack	An HTTP parser attack is an attempt to cause an HTTP parser to crash, consume excessive resources, run slowly, run an attacker's code, or cause the web application to do anything beyond its intended design.

Table 11.1 *Types of attacks detected by attack signatures*

Attack type	Description
HTTP request smuggling attack	HTTP request smuggling sends a specially formatted HTTP request that might be parsed differently by the proxy system and by the final system, so the attacker can smuggle a request to one system without the other one being aware of it. This attack makes it possible to exploit other attacks such as session hijacking, cross-site scripting (XSS), and the ability to bypass web application firewall protection.
HTTP response splitting	HTTP response splitting occurs when an attempt is made to deliver a malicious response payload to an application user.
Information leakage	Information leakage is when a web site reveals sensitive data, such as developer comments or error messages, which may aid an attacker in exploiting the system.
Injection attempt	An injection attempt is an attempt to include in a request information that is not permitted by the security policy, such as including a null in a request or including an illegal attachment.
LDAP injection	LDAP injection is an attack technique used to exploit web sites that construct LDAP statements from user-supplied input.
Malicious file upload	A malicious file upload refers to an attempt to upload a file that could cause damage to the system, for example, through the use of remote code execution or hostile data uploads.
Non-browser client	Non-browser client is an attempt by automated client access to obtain sensitive information. HTML comments, error messages, source code, or accessible files may contain sensitive information.
Other application attacks	This attack category represents attacks that do not fit into the more explicit attack classifications, including email injection, HTTP header injection, attempts to access local files, potential worm attacks, CDATA injection, and session fixation.
Other application activity	This attack category represents attacks that do not fit into the more explicit attack classifications.
Parameter tampering	Parameter tampering attacks involve the manipulation of parameters exchanged between client and server to modify application data, such as user credentials and permissions, or the price and quantity of products.
Path traversal	The path traversal attack technique forces access to files, directories, and commands that potentially reside outside the web document **root** directory.
Predictable resource location	Predictable resource location is an attack technique used to uncover hidden web site content and functionality.
Remote file include	Remote file include attacks occur as a result of unclassified application attacks such as when applications use parameters to pass URLs between pages.
Server-side code injection	SSI injection (server-side include) is a server-side exploitation technique that allows an attacker to send code into a web application, which is then run locally by the web server.

Table 11.1 *Types of attacks detected by attack signatures (Continued)*

Attack type	Description
Session hijacking	Web servers often send session tokens to the client browser upon successful client authentication. A session token is usually a string of variable width, and it could be placed in the URL, in the header of an HTTP request as a cookie, in other parts of the header of an HTTP request, or in the body of the HTTP request. Session hijacking compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the web server.
SQL-Injection	SQL Injection is an attack technique used to exploit web sites that construct SQL statements from user-supplied input.
Trojan/Backdoor/Spyware	Attackers use Trojan horse, backdoor, and spyware attacks to try to circumvent a web server's or web application's built-in security by masking the attack within a legitimate communication. For example, an attacker may include an attack in an email or Microsoft® Word document, and when a user opens the email or document, the attack launches.
Vulnerability scan	A vulnerability scan is an attack technique that uses an automated security program to probe a web application for software vulnerabilities.
Web scraping	Web scraping is the process of collecting information from web sites, typically using automated programs, or bots (short for web robots).
XML parser attack	An XML parser attack is an attempt to cause an XML parser to crash, consume excessive resources, run slowly, run an attacker's code, or cause the web application to do anything beyond its intended design.
XPath Injection	XPath injection attacks occur when an attempt is made to inject XPath queries to the vulnerable web application.

Table 11.1 *Types of attacks detected by attack signatures (Continued)*

Managing the attack signatures pool

The attack signatures pool contains all of the attack signatures that are part of the configuration. The pool includes the system-supplied attack signatures, which are the attack signatures that are shipped with the Application Security Manager, and any user-defined attack signatures. You can perform the following tasks to manage and maintain the attack signatures pool:

- Filter the attack signatures pool based on predefined or user-defined criteria. See *Working with the attack signatures pool filter*, following.
- View detailed information for the individual attack signatures. See *Viewing attack signature details*, on page 11-8.
- Update the system-supplied attack signatures. See *Updating the system-supplied attack signatures*, on page 11-10.
- Create a user-defined attack signature. See *Creating a user-defined attack signature*, on page 11-26.
- Import user-defined attack signatures. See *Importing user-defined attack signatures*, on page 11-28.
- Export user-defined attack signatures. See *Exporting user-defined attack signatures*, on page 11-29.

Working with the attack signatures pool filter

The attack signatures pool is quite large, so there is a filter that you can use to display only those signatures that you are interested in viewing. The filter has several built-in filter options. You can also create a custom filter.

Using the built-in filter options for attack signatures

The built-in filter options reduce the viewable attack signatures to a subset that matches a specific characteristic of the attack signatures. Table 11.2 describes the built-in filters.

Attack signatures built-in filter option	Description
Show all signatures	Use this built-in filter to display all attack signatures in the database.
Show signatures by name	Use this built-in filter to display signatures that match the name you provide.
Show signatures of accuracy greater than/equal to	Use this built-in filter to display only signatures whose accuracy is rated greater than or equal to the accuracy that you select. The ***attack signature accuracy*** indicates the ability of the attack signature to identify the attack, including susceptibility to false-positive alarms.

Table 11.2 *Built-in filter options for viewing the attack signatures pool*

Attack signatures built-in filter option	Description
Show signatures of risk greater than/equal to	Use this built-in filter to display only signatures whose risk is rated greater than or equal to the accuracy that you select. The ***attack signature risk*** indicates the level of potential damage this attack may cause, if it were successful.
Show signatures of attack type	Use this built-in filter to display only signatures that match the attack type that you select.

Table 11.2 *Built-in filter options for viewing the attack signatures pool (Continued)*

To view the attack signatures pool with a built-in filter

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens, where you can review the entire attack signatures pool.
2. From the **Filter** list, select a built-in filter.
   The screen refreshes, and displays either a text box or a select list for the selected filter.
3. Provide the appropriate information, and click the **Go** button.
   The screen refreshes, and displays only those attack signatures that match the criteria you selected.

Creating a custom filter for attack signatures

If the built-in filter options are too broad in scope, you can configure a custom filter option to view signatures in the attack signatures pool. For example, you can create a custom filter that displays attack signatures that apply only to parameters, or you can create a custom filter that displays only attack signatures for a specific attack type. When you create a custom filter, you can use one or more of the filter options, as required. Table 11.3 describes the custom filter options.

Attack signature custom filter option	Description
Containing String	Displays only attack signatures that contain the specified alpha-numeric string.
Signature ID	Displays only attack signatures that match a specific signature ID number. Signature ID numbers are system-supplied, and cannot be modified.
Apply To	Displays only attack signatures that apply either to requests or to responses.
Apply to Parameter	Displays only attack signatures that apply to parameters.
Apply to XML	Displays only attack signatures that apply to XML.

Table 11.3 *Custom filter options for the attack signatures pool*

Attack signature custom filter option	Description
Attack Type	Displays only attack signatures that match the selected attack type. See Table 11.1, on page 11-3, for a description of the attack types having signatures associated with them.
Systems	Displays only attack signatures that match the assigned systems.
Accuracy	Displays only attack signatures that match the criteria you select.
Risk	Displays only attack signatures that match the criteria you select.
User-defined	Displays only attack signatures that are user-defined.
Update Date	Displays only attack signatures that have been updated within the time frame you specify.

Table 11.3 *Custom filter options for the attack signatures pool (Continued)*

Viewing attack signature details

When you click the name of each attack signature, the system displays the properties listed in Table 11.4.

Property	Description
Name	Displays the signature name.
ID	Specifies the signature number automatically provided by the system.
Apply To	Indicates whether the rule inspects the client's request (Request) or the server's response (Response).
Systems	Displays which systems (for example web applications, web servers databases, and application frameworks) the signature protects.
Attack type	Displays the threat classification to which the attack signature applies. See *Types of attacks that attack signatures detect*, on page 11-3, for information on the specific types.
Accuracy	Indicates the ability of the attack signature to identify the attack including susceptibility to false-positive alarms: **Low**: Indicates a high likelihood of false positives. **Medium**: Indicates some likelihood of false positives. **High**: Indicates a low likelihood of false positives.
Risk	Indicates the level of potential damage this attack might cause if it is successful: **Low**: Indicates the attack does not cause direct damage or reveal highly sensitive data. **Medium**: Indicates the attack may reveal sensitive data or cause moderate damage. **High**: Indicates the attack may cause a full system compromise.

Table 11.4 *Signature properties*

Property	Description
User-defined	Indicates whether this signature is a system supplied rule (**No**) or was defined by a user (**Yes**).
Revision	Indicates the version of the attack signature.
Last Updated	Indicates the date when the attack signature was most recently updated.
Documentation	Indicates whether the system provides documentation explaining this attack signature (**View**) or not (**N/A**). Click the **View** link to display the available documentation.
References	Displays a clickable link to an external web site explaining this attack signature, or displays (**N/A**) if no link is available.

Table 11.4 *Signature properties (Continued)*

To view attack signature details

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. In the Signature Name column, click (the Show/Hide Details icon) next to the signature for which you want to view information.
   Basic information display on the screen below the signature.

   Tip*: You can also click the name of the signature to display the signature properties.*
3. For the **Documentation** setting, click **View** to see additional information that applies to the selected attack signature.
   A new screen opens (Documentation for Attack Signature), and displays the additional documentation.

   Note*: Some attack signatures have no additional documentation. You see **N/A** for the **Documentation** setting if this is the case.*
4. When you finish reviewing the documentation, click the **Close** button.
   The Attack Signatures screen reopens.

Updating the system-supplied attack signatures

You can update the system-supplied attack signatures on a regular basis to ensure that your applications are protected against new attacks and threats.

When you update the system-supplied attack signatures, the update includes any new signatures that are available, and also updates any existing attack signatures that have been revised, including the signature documentation. You can configure automatic updates, or you can update them manually.

Important considerations when updating attack signatures

Two conditions may cause an attack signature update to fail: insufficient network access and duplicate attack signature names. In addition, it is a good idea to update the attack signatures on all Application Security Manager systems, keeping them in sync.

Ensuring network access

The Application Security Manager must have external network access for the update process to work. To obtain the updated signature files, you must also have both a valid service agreement with F5 Networks, and a service check date within 18 months of the signature-file update request.

If your license has lapsed, you must re-license the Application Security Manager. If you need details about allowing signature file updates through a firewall or an HTTPS proxy, refer to Solution 8217, *Updating the BIG-IP ASM attack signatures*, on the F5 Networks Technical Support web site. Contact F5 Networks Technical Support, **https://support.f5.com,** for additional assistance.

Resolving name duplication issues in user-defined attack signature names

Do not use system-supplied attack signature names when you create a user-defined attack signature. Although the system does not prohibit duplicate attack signature names, future attack-signature updates may fail because of name conflicts.

If you inadvertently duplicate a system-supplied attack signature name, rename the user-defined attack signature (see *Modifying a user-defined attack signature*, on page 11-27, for more information). You can then retry the update process.

Keeping attack signatures in sync

F5 Networks recommends that you have the same set of attack signatures on all Application Security Manager systems, especially if you plan to copy security policies from one system to another. If you are updating the attack signatures on one system, it is a good idea to update them on all of the systems, to maintain synchronization and consistency with system security.

Configuring automatic updates for system-supplied attack signatures

You can configure automatic updates so that the system updates the attack signatures database on a regular schedule.

To automatically update system-supplied attack signatures

1. In the navigation pane, expand **Application Security**, point to **Options**, **Attack Signatures**, and then click **Attack Signatures Update**.
   The Attack Signatures Update screen opens.
2. For the **Update Mode** setting, click **Scheduled**.
3. For the **Update Interval**, select how often you want the system to check for updates (daily, weekly, or monthly).
4. If you want the system to automatically apply all active security policies after updating the system-supplied signatures, leave the **Auto Apply Policy After Update** box checked.
5. Click the **Save Settings** button to preserve any changes you may have made to the configuration.

Configuring manual updates for system-supplied attack signatures

If you want to update the system-supplied attack signatures manually, you can use the manual update option. You can obtain the latest attack signatures update file from **http://downloads.f5.com**.

To manually update system-supplied attack signatures

1. In the navigation pane, expand **Application Security**, point to **Options**, point to **Attack Signatures**, and then click **Attack Signatures Update**.
   The Attack Signatures Update screen opens.
2. In the Attack Signatures Update area, for the **Update Mode** setting, click **Manual**.
   The screen refreshes, and displays the **Delivery Mode** setting.
3. For the **Delivery Mode** setting, select one of the following options:
   - Select **Automatic** if you want to go directly to the web server for the latest update file.
   - Select **Manual** if you want the system to save the updates in a file that you can apply at a later time. The screen displays the **Upload File** setting, where you can specify the path to the file that contains the updates.
4. If you want the system to automatically apply the new attack signatures configuration to all active security policies, leave the **Auto Apply Policy After Update** box checked.

5. Click the **Save Settings** to save your changes.
6. Click the **Update Signatures** button to start the update process.

Viewing information about the most recent update

The Application Security Manager records the logistical information about the most recent update activity, and displays this information on the Attack Signatures Update screen. You can review the last update time, as well as the readme file that pertains to the update.

To review information about the most recent update

1. In the navigation pane, expand **Application Security**, point to **Options**, point to **Attack Signatures**, and then click **Attack Signatures Update**.
   The Attack Signatures Update screen opens.
2. In the Latest Update Details area, you can review the creation date and time for the database, as well as the date and time at which the database was most recently updated.
3. For the **Readme** option, click **View Readme** to see the details regarding the update.

Receiving email notification of attack signature updates

If you want to receive notification from F5 Networks about attack signature updates available for download, you can sign up on the Ask F5^{SM} web site for the Security email distribution list. Once you are on the distribution list, F5 Networks sends an email message whenever attack signature updates are available.

Note

You must have a valid service contract, and an Ask F5^{SM} account, to receive the attack signature update notifications.

To sign up for the Security email distribution list

1. Open a browser, and log in to the Ask F5^{SM} web site at **https://support.f5.com**.
   The Ask F5 Knowledge Base screen opens.
2. In the navigation pane, click the **Mailing Lists** button.
   The TECHNEWS screen opens.
3. In the Security area, click the **security-subscribe@lists.f5.com** link.
4. Send the blank email message, as is.
   The list manager adds your email address to the Security email distribution list.

Working with attack signature sets

Rather than assigning individual attack signatures to a security policy, you assign attack signature sets. By default, when you create a new security policy, the system automatically assigns the Generic Detection Signatures set to the security policy. In addition to the generic signatures set, you can assign one of the other system-supplied signatures sets to the security policy, and you can create a signature set and assign that to the security policy. You can also remove all signature set assignments from a security policy, although F5 Networks recommends against doing this.

When you create an attack signature set, you can tailor the attack signatures to your specific systems and applications. For more information on assigning an attack signature set to a security policy, see *Assigning attack signature sets to a security policy*, on page 11-17.

Viewing system-supplied signature sets

The Application Security Manager ships with several system-supplied signature sets. By default, the Generic Detection Signatures system-supplied set is associated with all new security policies. Table 11.5 describes the system-supplied signature sets.

System-supplied signature set	Description
Generic Detection Signatures	Targets well-known or common web and application attacks.
OWA Signatures	Targets attacks against the Microsoft® Outlook Web Access (OWA) application.
WebSphere Signatures	Targets attacks on a variety of different computing platforms integrated using WebSphere including general database, Microsoft Windows, IIS, Microsoft SQL Server, Apache, Oracle, Unix/Linux, IBM DB2, PostgreSQL, and XML.
Low Accuracy Signatures	Contains signatures that have a low level of accuracy and produce more false positives when identifying attacks.
Medium Accuracy Signatures	Contains signatures that have a medium level of accuracy when identifying attacks.
High Accuracy Signatures	Contains signatures that have a high level of accuracy and produce few false positives when identifying attacks.
All Signatures	Contains all of the attack signatures in the attack signature pool.

Table 11.5 *System-supplied attack signature sets*

Creating an attack signature set

You can create signature sets in two ways: by using a filter or by manually selecting the signatures to include. Filter-based signature sets are based solely on criteria you define in the signatures filter. The advantage to filter-based signature sets is that you can focus on the criteria that define the attack signatures you are interested in, rather than trying to manage a specific list of attack signatures. Another advantage to filter-based sets is that when you update the attack signatures database, the system also updates any signature sets to which the update applies.

To create a filter-based attack signature set

1. In the navigation pane, expand **Application Security**, point to **Options**, point to **Attack Signatures**, and then click **Attack Signature Sets**.
   The Attack Signature Sets screen opens.
2. Above the Attack Signature Sets list, click **Create**.
   The Create Signature Set screen opens.
3. In the Create Signature Set area, in the **Name** box, type a unique name for the signature set.
4. For the **Type** setting, select **Filter-based**.
5. For the **Default Blocking Actions** setting, decide which blocking actions you want the system to enforce for the signature set when you associate it with a new security policy.

   ***Note**: The **Learn**, **Alarm**, and **Block** settings take effect only when you assign this signature set to a new security policy. If this signature set is already assigned to an existing security policy, these settings have no effect.*
6. If you want the system to automatically include this signature set in any new security policies you create, check the **Assign To Policy By Default** box.
7. In the Signatures Filter area, select the filter options you want to use to create the new signature set. For descriptions of the individual filter options, see the online help.
8. In the Signatures area, for the **Signatures** setting, you can review the signatures list that the filter settings generate.
   The list content changes dynamically with the filter selection.
9. Click the **Create** button.
   The screen refreshes, and you see the new signature set in the Signatures Set list.
10. Associate the signature set with security policies, as needed. See *Assigning attack signature sets to a security policy*, on page 11-17.

Manual signature sets are composed of attack signatures that you individually select from the attack signatures pool. You can use the signatures filter to help narrow the scope of the available signatures in the pool, however, once the manual signature set is created, the system does not retain the filter criteria.

To create a manual attack signature set

1. In the navigation pane, expand **Application Security**, point to **Options**, **Attack Signatures**, and then click **Attack Signature Sets**.
   The Attack Signature Sets screen opens.
2. Above the Attack Signature Sets list, click **Create**.
   The Create Signature Set screen opens.
3. In the Create Signature Set area, in the **Name** box, type a unique name for the signature set.
4. For the **Type** setting, select **Manual**.
5. For the **Default Blocking Actions** setting, decide which blocking actions you want the system to enforce for the signature set when you associate it with a new security policy and enable them.

   ***Note**: The **Learn**, **Alarm**, and **Block** settings take effect only when you assign this signature set to a new security policy. If this signature set is already assigned to an existing security policy, these settings have no effect.*
6. If you want the system to automatically include this signature set in any new security policies you create, check the **Assign To Policy By Default** box.
7. In the Signatures Filter area, use the filter options to reduce the scope of the **Available signatures** list (in the Signatures area). For descriptions of the individual filter options, see the online help.
   The list content changes dynamically with the filter selection.
8. For the **Signatures** setting, move the signatures you want to include in the set into the assigned signatures list.
9. Click the **Create** button.
   The screen refreshes, and you see the new signature set in the Signatures Set list.
10. Associate the signature set with security policies, as needed. See *Assigning attack signature sets to a security policy*, on page 11-17.

Editing used-defined attack signature sets

You can edit user-defined attack signature sets to add or remove signatures, or change the properties of the signature set. When you edit attack signature sets, the changes apply to all of the security policies to which the set is assigned. Additionally, filter-based signature sets automatically receive any applicable updates when you use the attack signature update feature. (For more information, see *Updating the system-supplied attack signatures*, on page 11-10.)

To edit an attack signature set

1. In the navigation pane, expand **Application Security**, point to **Options**, **Attack Signatures**, and then click **Attack Signature Sets**.
   The Attack Signature Sets screen opens.
2. In the Name column, click the name of the signature set that you want to edit.
   The Signature Set Properties screen opens.
3. Make any changes as required.
4. Click the **Save** button below the Signatures area.
   The system updates the signature set in all security policies that include this set.

Deleting a user-defined attack signature set

You can remove a user-defined signature set from the configuration. When you delete a signature set, you are not deleting the attack signatures that make up the set. You are, however, removing the signature set from the security policy, which may have significant ramifications on the security policy's effectiveness.

Note

You cannot delete system-supplied attack signature sets.

To delete an attack signature set

1. In the navigation pane, expand **Application Security**, point to **Options**, **Attack Signatures**, and then click **Attack Signature Sets**.
   The Attack Signature Sets screen opens.
2. Select the check box preceding the signature set that you want to remove, and click the **Delete** button.
   A confirmation popup screen displays.
3. Click **OK**.
   The system removes the selected signature set from the configuration.

Assigning attack signature sets to a security policy

Each security policy has its own attack signature sets. By default, the system assigns the **Generic Attack Signatures** to all security policies. In additions, you can assign any additional attack signature sets to a security policy, including any system-supplied set, or those that you may have created.

To assign an attack signature set to a security policy

1. In the navigation pane, expand **Application Security**, click **Attack Signatures**.
   The Attack Signature Sets Assignment screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. In the Attack Signature Sets Assignment area, in **Available Signature Sets** list, select the attack signature sets that you want to assign to the security policy.

   ***Tip**: To select more than one set, hold the Ctrl key and click the names.*
4. Move the attack signature sets that you want included in the security policy into the **Assigned Signature Sets** list.
5. Click the **Update** button to save any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Viewing the attack signature sets for a specific security policy

You can review the attack signature sets that are associated with a security policy from the Signature Sets screen. By default, the system assigns the signature set, **Generic Detection Signatures**, to all new security policies. Additionally, the system includes in the security policy any attack signature sets you selected with the Deployment wizard.

To view attack signature sets for a specific security policy

1. In the navigation pane, expand **Application Security**, click **Attack Signatures**.
   The Attack Signature Sets Assignment screen opens.
2. In the editing context area, ensure that the edited security policy is the one whose attack signature sets you want to view.
3. In the **Assigned Signature Sets** list, you can review the signature sets that are associated with the security policy, as well as the blocking policy actions for signatures in the set.

◆ Tip

Click a signature set name to review the attack signatures in that set.

Viewing all attack signatures for a security policy

When you assign an attack signature set to a security policy, the system builds a list of all of the attack signatures. You can review the signatures, their current blocking policy, and their state on the Policy Attack Signatures screen. Figure 11.1 shows a sample of the attack signature list for a security policy.

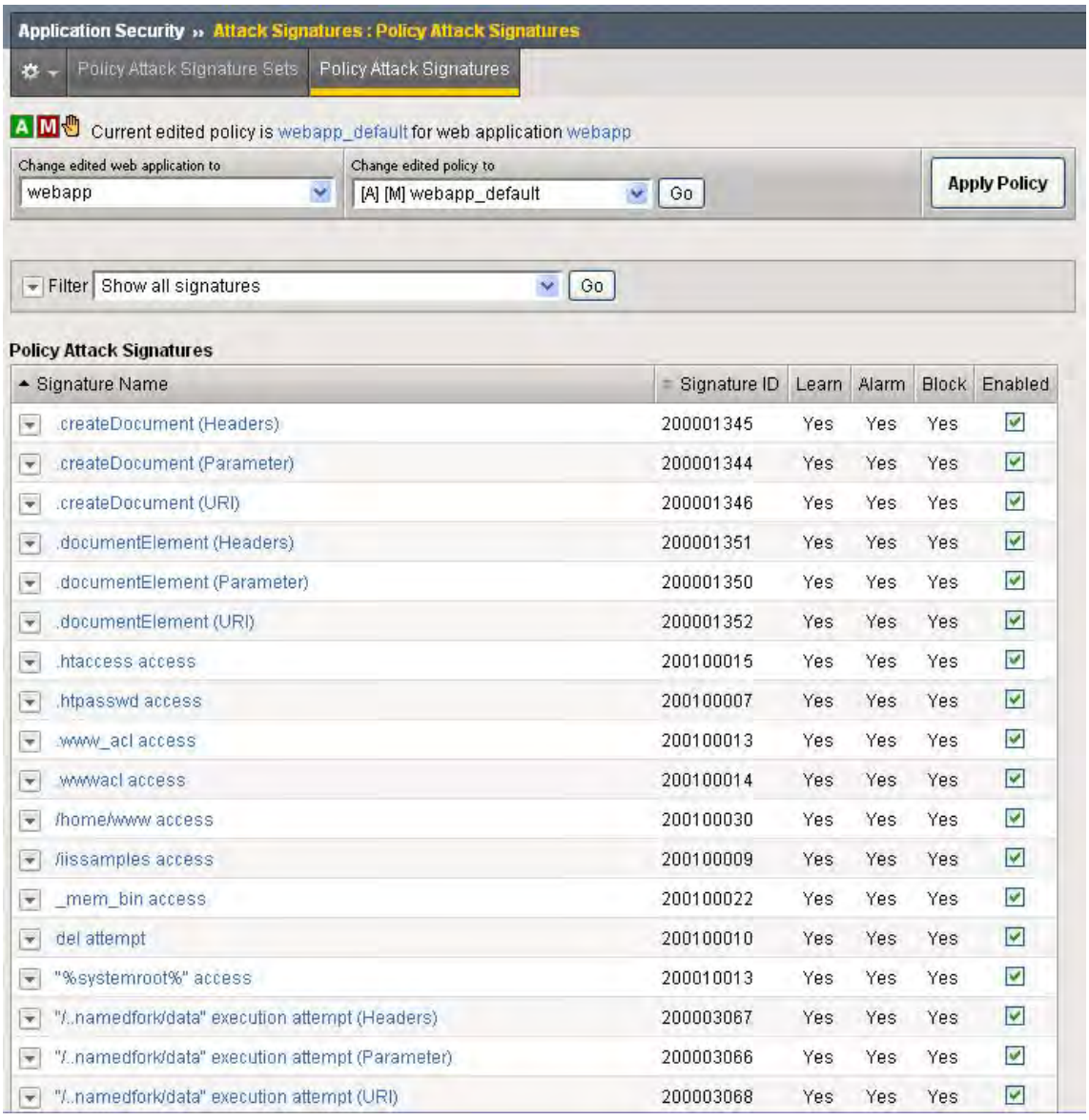

Application Security » Attack Signatures : Policy Attack Signatures
Policy Attack Signature Sets
Policy Attack Signatures
Current edited policy is webapp_default for web application webapp
Change edited web application to
webapp
Change edited policy to
[A] [M] webapp_default
Go
Apply Policy
Filter
Show all signatures
Go
Policy Attack Signatures
Signature Name
Signature ID
Learn
Alarm
Block
Enabled
.createDocument (Headers) 200001345 Yes Yes Yes
.createDocument (Parameter) 200001344 Yes Yes Yes
.createDocument (URI) 200001346 Yes Yes Yes
.documentElement (Headers) 200001351 Yes Yes Yes
.documentElement (Parameter) 200001350 Yes Yes Yes
.documentElement (URI) 200001352 Yes Yes Yes
.htaccess access 200100015 Yes Yes Yes
.htpasswd access 200100007 Yes Yes Yes
.www_acl access 200100013 Yes Yes Yes
.wwwacl access 200100014 Yes Yes Yes
/home/www access 200100030 Yes Yes Yes
/iissamples access 200100009 Yes Yes Yes
_mem_bin access 200100022 Yes Yes Yes
del attempt 200100010 Yes Yes Yes
"%systemroot%" access 200010013 Yes Yes Yes
"/..namedfork/data" execution attempt (Headers) 200003067 Yes Yes Yes
"/..namedfork/data" execution attempt (Parameter) 200003066 Yes Yes Yes
"/..namedfork/data" execution attempt (URI) 200003068 Yes Yes Yes

Figure 11.1 *Attack signatures associated with a security policy*

To view all attack signatures for a security policy

1. In the navigation pane, expand **Application Security**, point to **Attack Signatures**, and then click **Policy Attack Signatures**. The Policy Attack Signatures screen opens.
2. In the editing context area, ensure that the edited security policy is the one whose attack signatures you want to view.
3. For the **Filter** option, select **Show all signatures** to display all signatures associated with the security policy.

Disabling an attack signature in a security policy

You can disable attack signatures in a security policy, one at a time.

To disable specific attack signatures in a security policy

1. In the navigation pane, expand **Application Security**, point to **Attack Signatures**, and then click **Policy Attack Signatures**. The Policy Attack Signatures screen opens.
2. In the editing context area, ensure that the edited security policy is the one with attack signatures you want to disable.
3. For the **Filter** option, select **Show all signatures** to display all signatures associated with the security policy.
4. In the Policy Attack Signatures list, clear the **Enabled** box next to each signature you want to disable.
5. Click **Save**.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Modifying the blocking policy for an attack signature set

The blocking policy defines how the Security Enforcer processes requests that trigger violations. For each attack signature set that is assigned to a security policy, you enable one or more of the blocking actions:

- Learn
- Alarm
- Block

For more information on the Blocking Policy and the blocking actions, refer to *Configuring security policy blocking*, on page 6-41.

When the signatures have passed the staging period and before the system applies the blocking actions, you have a chance to review the attack signatures list and decide which ones to enable or disable. For information on how to do this, refer to *Enabling or disabling signatures in staging*, on page 11-23.

Note

The blocking policy applies to all of the signatures in the signature set. You cannot specify a blocking policy for individual signatures.

To configure the blocking actions for an attack signature set

1. In the navigation pane, expand **Application Security** and click **Attack Signatures**.
   The Attack Signature Sets Assignment screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want update.
3. In the Attack Signature Sets Assignment area, for each signature set in the **Assigned Signature Sets** list, check or clear the boxes in the Learn, Alarm, and Block columns as required.

   ***Note**: If the enforcement mode for the security policy is transparent, the Block action is not configurable. You can enable or disable the Block action only when the enforcement mode is blocking.*
4. Click the **Update** button to save any changes you may have made.
5. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Understanding attack signature staging

When you first activate a security policy, the system puts the attack signatures into staging (if staging is enabled). ***Staging*** means that the system applies the attack signatures to the web application traffic, but does not apply the blocking policy block action to requests that trigger those attack signatures. The default staging period is seven days. Whenever you add or change signatures in assigned sets, those are also put into staging. (For more information on updating signatures, see *Updating the system-supplied attack signatures*, on page 11-10.)

To modify the staging configuration

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. In the editing context area, ensure that the edited security policy is the one you want to update.
3. For the **Staging-Tightening Period** setting, type the number of days for which you want new or updated attack signatures to remain in staging.
4. Check the **Enable Signature Staging** box.
5. Click **Save** to retain any changes you may have made.
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Managing signatures in staging that generate learning suggestions

Placing new and updated attack signatures in staging helps to reduce the number of violations triggered by false-positive matches. When signatures match attack patterns during the staging period, the system generates learning suggestions. You can view these attack signatures from the Attack Signature Staging screen. Upon evaluation, if the signature is a false-positive, you can disable the signature, and the Security Enforcer no longer applies that signature to traffic for the corresponding web application. Alternately, if the detected signature match is legitimate, you can enable the corresponding attack signature. Note that enabling the signature removes it from staging, and puts the blocking policy into effect.

To view signatures in staging that generate learning suggestions

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the Traffic Learning area of the Negative Security Violations setting, click the **Attack signature staging** link.
   The Attack Signature Staging screen opens, where you can review the signatures that detected attack patterns in a request.

Figure 11.2 shows the **Attack signature staging** link on the Traffic Learning screen.

Figure 11.2 *Link to attack signatures in staging*

Figure 11.3 shows a sample screen with examples of the attack signatures that are in staging for the current edited security policy. On your screen, click the number under Recent Incidents to view details about requests that caused violation for that signature.

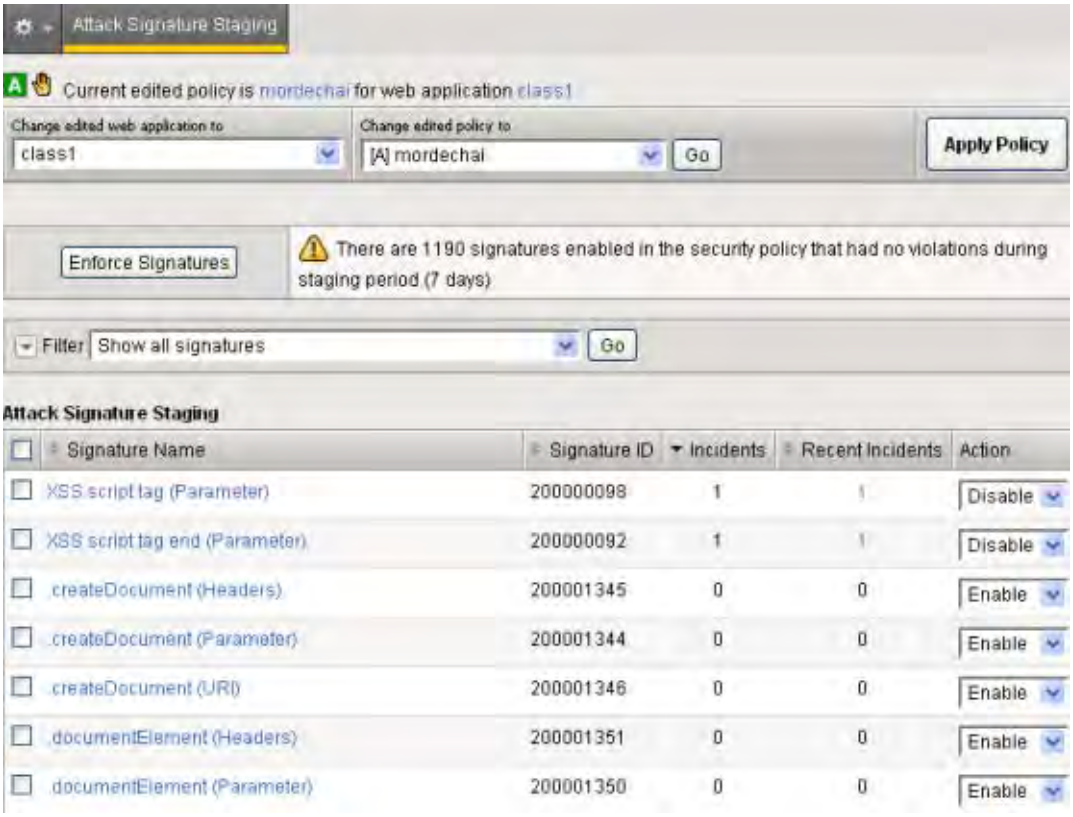

Attack Signature Staging
Current edited policy is mordechai for web application class1
Change edited web application to
class1
Change edited policy to
[A] mordechai
Go
Apply Policy
Enforce Signatures
There are 1190 signatures enabled in the security policy that had no violations during staging period (7 days)
Filter
Show all signatures
Go
Attack Signature Staging
Signature Name
Signature ID
Incidents
Recent Incidents
Action
XSS script tag (Parameter) 200000098 1 1 Disable
XSS script tag end (Parameter) 200000092 1 1 Disable
createDocument (Headers) 200001345 0 0 Enable
createDocument (Parameter) 200001344 0 0 Enable
createDocument (URI) 200001346 0 0 Enable
documentElement (Headers) 200001351 0 0 Enable
documentElement (Parameter) 200001350 0 0 Enable

Figure 11.3 *Attack signatures in staging*

Enabling or disabling signatures in staging

If a new or updated attack signature in staging detects an attack pattern in the web application traffic, you should review the signature details and the requests that triggered the attack signature. If the detected attack pattern is not an actual threat, the signature has generated a false-positive alarm. If a particular attack signature triggers false-positives, you may want to disable that particular attack signature.

In some situations, you may want to take action to enable or disable an attack signature immediately, rather than wait for the staging period to complete. For example, if a signature detects a legitimate attack pattern, you may want to enable that signature right away, instead of waiting for the staging period to end.

Another example is when a trusted-traffic signature match is detected but the request is legitimate. In such a case, you should disable that signature immediately.

You can configure the system to log all of the attack signatures that you disabled. For details, see the **LogSignatures** parameter in Appendix D, *Internal Parameters for Advanced Configuration*.

To disable or enable an attack signature in staging

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the Negative Security Violations section of the Traffic Learning area, click the **Attack signature staging** link.
   The Attack Signature Staging screen opens.
3. To view the signature details, click the signature name.
4. To view the requests in which the signature detected an attack pattern, click the number.
5. For each signature that you want to enable or disable, perform the following tasks:
   a) In the Action column, select **Enable** or **Disable** from the list.
   b) In the Select column (far left), check the select box next to the signature name.

   ***Tip**: To disable a signature for a specific parameter when at least one incident occurred, select the **Disable on parameter** action.*
6. Below the Attack Signature Staging area, click the **Apply** button.
   A confirmation popup screen opens.
7. Click **OK**.
   The popup screen closes, and displays the Traffic Learning screen.
8. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

Enforcing all attack signatures

After the staging period is over, you can enforce all attack signatures that did not cause violations, all at once.

To enforce all signatures

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the Negative Security Violations section of the Traffic Learning area, click the **Attack signature staging** link.
   The Attack Signature Staging screen opens.
3. Click the **Enforce Signatures** button.
   The system removes from staging all signatures that did not cause violations and enforces them.

Managing user-defined attack signatures

User-defined attack signatures are those that the user creates and adds to the attack signature pool. User-defined attack signatures have the following attributes:

- They adhere to the rule syntax as explained in Appendix C, *Syntax for Creating User-Defined Attack Signatures*.
- They may, but are not required to, contain any of the properties of the system-supplied signatures.
- They are never updated by F5 Networks. All user-defined signatures are carried forward as-is whenever there is a software version upgrade.
- They are placed in staging whenever a user adds or changes any of the signature properties. (See *Understanding attack signature staging*, on page 11-21, for more information.)

Creating a user-defined attack signature

You can create a user-defined attack signature rule using the syntax that is explained in Appendix C, *Syntax for Creating User-Defined Attack Signatures*.

To create a user-defined attack signature

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. Above the Attack Signatures list, click **Create**.
   The Create Attack Signature screen opens.
3. In the **Name** box, type a unique name for the new attack signature.

   Warning*: If you create a user-defined attack signature with the same name as a system-supplied attack signature, subsequent signature updates may fail.*
4. In the **Description** box, type an optional description of the signature.
5. For the **Apply To** setting, select whether the new signature applies to requests or responses.
6. For the **Systems** setting, select from the **Available Systems** list any systems to which the new signature applies, and use the Move buttons (<< and >>) to add them to the **Assigned Systems** list.
7. For the **Attack Type** setting, select the type of threat that the new signature protects against.
8. For the **Rule** setting, type a rule according to the syntax guidelines in Appendix C, *Syntax for Creating User-Defined Attack Signatures*. This setting is required.
9. For the **Accuracy** setting, select an accuracy level. The accuracy level indicates the ability of the attack signature to identify the attack, including susceptibility to false-positive alarms.
10. For the **Risk** setting, select a risk level. The risk level indicates the level of potential damage this attack may cause, if it were successful.
11. Click the **Create** button.
    The screen refreshes, and displays the Attack Signatures list.

Modifying a user-defined attack signature

You may need to update a user-defined attack signature, for example, to change the accuracy level after the signature has been in use for a while, based on observed traffic.

To modify a user-defined attack signature

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. In the Attack Signatures list, click the name of the user-defined attack signature that you want to modify.
   The Update Attack Signature screen opens.
3. Make changes to the attack signature, as needed.
4. Click **Save** to retain the changes.

Deleting a user-defined attack signature

You can permanently remove user-defined attack signatures from the attack signature pool. Note that when you delete a user-defined signature from the attack signature pool, the system removes that signature from any signature sets of which the attack signature is a member.

To delete a user-defined attack signature

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. In the Attack Signatures list, in the Select column (far left), check the Select box next name of the user-defined attack signature that you want to delete.
3. Below the Attack Signatures list, click the **Delete** button.
   A confirmation popup screen displays.
4. Click the **OK** button.
   The system deletes the attack signature from the configuration, and displays the Attack Signatures list screen.

Importing user-defined attack signatures

If you have a large number of user-defined attack signatures that you want to add to the configuration, you can import them in an XML-formatted file. You can also use the import functionality to import a previously exported user-defined attack signature file from the same version of Application Security Manager. Figure 11.4 shows an example of the XML file format for the user-defined attack signatures file.

Note

The XML file format is the only accepted import format for attack signatures.

```
<?xml version="1.0" encoding="utf-8"?>
<signatures export_version="10.0.0">
  <sig>
    <rev>
      <sig_name>Unique signature name</sig_name>
      <rule>msg:"Signature Name"; content:"foo";</rule>
      <last_update>2007-03-26 10:20:33</last_update>
      <apply_to>Request</apply_to>
      <risk>3</risk>
      <accuracy>2</accuracy>
      <doc>Any additional descriptive text</doc>
      <attack_type>Cross Site Scripting (XSS)</attack_type>
      <systems>
        <system_name>IIS</system_name>
        <system_name>Microsoft Windows</system_name>
      </systems>
    </rev>
  </sig>
</signatures>
```

Figure 11.4 *XML format for user-defined attack signatures file*

WARNING

*The **sig_name** attribute uniquely identifies a user-defined attack signature. Therefore, when you import an attack signature XML file, if there are any signatures in the XML file whose **sig_name** attribute matches that of any existing user-defined signatures, the system overwrites the existing definition with the imported definition.*

To import a user-defined attack signature file

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. Above the Attack Signatures list, click **Import**.
   The Import Attack Signatures screen opens.

3. In the **Choose File** box, type the path to the XML file that contains the user-defined attack signatures. Alternately, click the **Browse** button and navigate to the XML file.
4. Click the **Import** button.
   The system imports the user-defined signatures, and issues either a success message or a failed message.
5. If the import is successful, click the **OK** button.
   The screen refreshes, and displays the Attack Signatures list with the additional user-defined signatures.
6. If the import was not successful, make any required changes to the XML file, and then try to import the file again.

Exporting user-defined attack signatures

You can export user-defined attack signatures to transfer them to another system, or save them in a remote location. When you export user-defined attack signatures, the Application Security Manager saves them in an XML file using the format shown in Figure 11.4, on page 11-28.

Note

You cannot export system-supplied attack signatures. You can export only user-defined attack signatures.

To export a user-defined attack signature file

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. Above the Attack Signatures list, click **Export**.
   The web browser opens a file download or a save file popup screen.

   ***Note**: Each web browser manages this functionality differently.*
3. Save the signature file in a location that meets your requirements. Application Security Manager saves the signature in a file name with this format:

   `sigfile_<date>_<time>.xml`
4. Close the popup screen.

12

Protecting XML Applications

- Getting started with XML security
- Configuring security for SOAP web services
- Implementing web services security
- Configuring security for XML content
- Fine-tuning XML defense configuration
- Masking sensitive XML data
- Associating an XML profile with a URL
- Associating an XML profile with a parameter
- Modifying XML security profiles

Getting started with XML security

Because XML is used as a data exchange mechanism, it is important to inspect, validate, and protect XML transactions. With XML security, you can protect the following applications:

- Web services that use HTTP as a transport layer for XML data
- Web services that use encryption and decryption in HTTP requests
- Web services that require verification and signing using digital signatures
- Web applications that use XML for client-server data communications, for example, Microsoft® Outlook Web Access

You implement XML security by creating an XML profile for a security policy. The XML profile can protect XML applications in the following ways:

- Enforces compliance against XML schema files or WSDL documents
- Validates XML format
- Implements defense rules for XML documents
- Masks sensitive XML data
- Encrypts and decrypts parts of SOAP (Simple Object Access Protocol) web services
- Signs and verifies parts of SOAP messages using digital signatures

Before you begin, you need the following information about the XML application that you want to protect:

- *Does the application use validation files, for example, an XML schema or WSDL document?*
  If yes, you must know which files and know where they are.
- *For web services, do the clients support secure web services with encryption and decryption capabilities?*
  If so, you can configure web services security to handle the decryption and encryption of XML data.
- *Does the application use XML digital signatures for signing and verification?*
  Web services security can verify requests and sign responses.
- *What applications are on the back end?*
  There can be more than one, for example, an Expat XML parser and an Oracle® database server.

You must have already created a security policy for a web application using the Deployment wizard by following the steps in *Creating a Security Policy for XML Transactions* in ***BIG-IP® Application Security Manager™: Getting Started Guide.***

How you proceed with configuring XML security depends on the type of application you want to protect:

- **For SOAP web services**: refer to *Configuring security for SOAP web services*, on page 12-3.
- **For XML content**: refer to *Configuring security for XML content*, on page 12-14.

Figure 12.1 shows an overview of the tasks for configuring XML security.

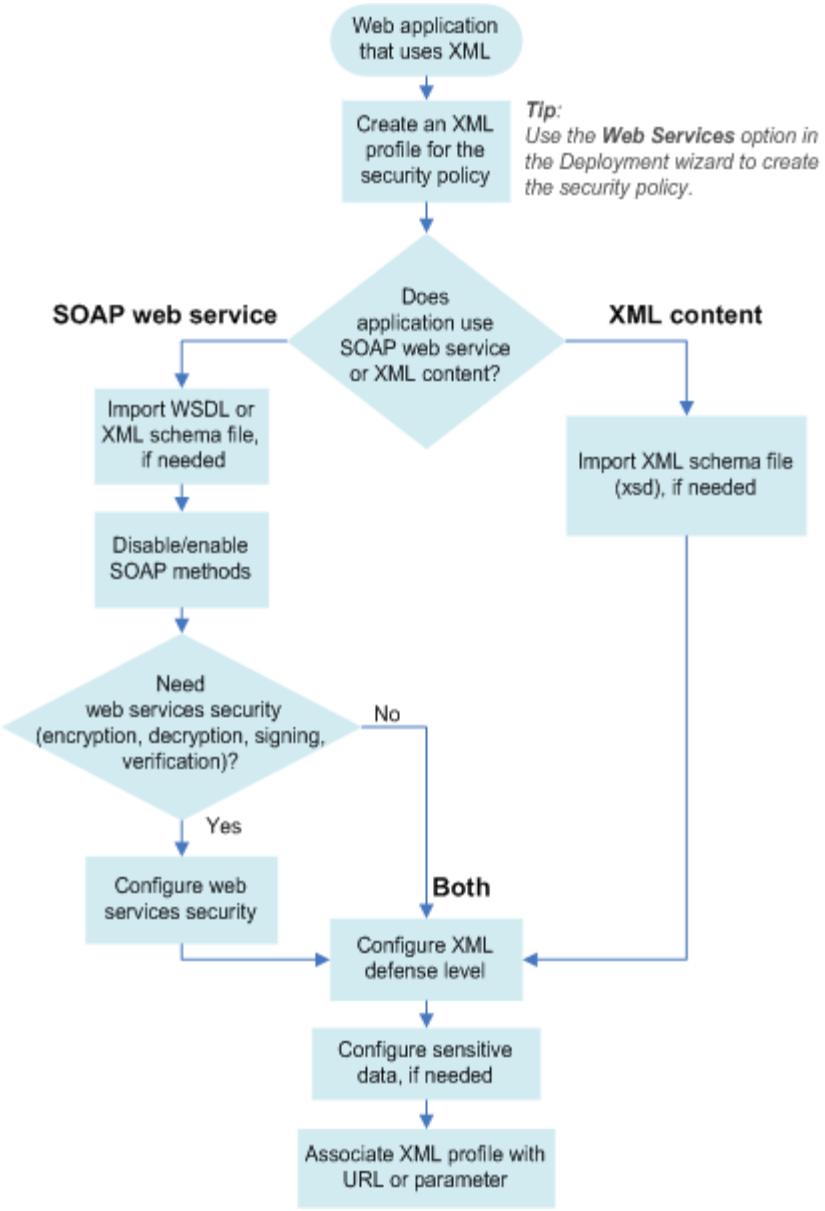

Web application that uses XML
Create an XML profile for the security policy
Tip:
Use the Web Services option in the Deployment wizard to create the security policy.
Does application use SOAP web service or XML content?
SOAP web service
XML content
Import WSDL or XML schema file, if needed
Disable/enable SOAP methods
Import XML schema file (xsd), if needed
Need web services security (encryption, decryption, signing, verification)?
No
Yes
Configure web services security
Both
Configure XML defense level
Configure sensitive data, if needed
Associate XML profile with URL or parameter

Figure 12.1 *Flowchart for configuring XML security*

Configuring security for SOAP web services

To configure security for SOAP web services, the security policy needs to have an XML profile. An ***XML profile*** defines the XML properties that a security policy enforces for XML applications. You must associate the XML profile with a URL or with a parameter.

Some web services have a WSDL or XML schema document to describe the language that the application uses to communicate with its remote users and systems. The XML profile can validate whether the incoming traffic complies with the WSDL or XML schema document. However, neither a WSDL nor an XML schema file is required for configuring security for web services.

Note

Creating an XML profile requires external network access to verify the XML schema link. The time needed to create an XML profile varies, depending on the size of the WSDL document or XML schema file, and your connection speed.

If you used the Deployment wizard to create a security policy by selecting the **Web Services (XML +WSDL/User Schema)** scenario, you already have a security policy with an XML profile. You can go to **XML Profiles** and click the profile you created to review its settings with the following procedure, or skip to *Implementing web services security*, on page 12-5 to configure encryption and signing.

To create an XML profile for web services security

1. In the navigation pane, expand **Application Security** and click the create icon (**)** next to **XML Profiles**.
   The Create New XML Profile screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the **Profile Name** box, type a name for the XML profile.
4. If you plan to implement web services security, check the **Enable Web Services Security** box, and refer to *Implementing web services security*, on page 12-5, for details about additional tasks that you should perform.

5. For the **Configuration Files** setting, if your web service uses a WSDL or XML schema file, perform steps a and b. Otherwise, skip to step 8.

   a) For the **File** option, click **Browse**, and navigate to the **.wsdl** or **.xsd** file.

   Note: *The file you upload must use UTF-8 character encoding.*

   b) Click **Upload**.
   The system uploads the file and lists its contents on the screen.

   Important*: When a WSDL or XML schema document refers to another WSDL or XML schema document, the system gives you the option of importing it. If circular dependencies exist in the files (for example, schema 1 references schema 2, which contains a reference to back to schema 1) import schema 1, then schema 2, then schema 1 again. This creates a mapping between the files.*

6. If you specified a referenced file type (in step 5), in the **Import URL** box, type:
   - For a WSDL file, the URL defined in the **location** directive
   - For an XSD file, the URL defined in the **schemaLocation** directive

7. To attempt to locate and use files referenced in the WSDL or XML schema document, ensure that the **Follow Schema Links** box is checked.

   To use this setting, make sure the DNS server is on the DNS lookup server list, and configure the DNS server on the BIG-IP system (**System>>Configuration>>Device>>DNS)**.

   Tip: *If you disable this setting and the uploaded file refers to other XML schemas, the system lists the referenced files in an error message at the top of the screen.*

8. To permit SOAP messages to contain attachments, check the **Allow Attachments in SOAP Messages** box.

9. If you imported a WSDL document as part of the configuration, perform these additional steps:

   a) For the system to verify the **SOAPAction** header, check the **Validate SOAPAction Header** box. The system automatically enables this setting when you upload a WSDL file.

   b) Review the **Valid SOAP Methods**; to disable any of them, clear the **Enabled** check box. For details, see *Managing SOAP methods*, on page 12-13.

10. In the Defense Configuration area, for **Defense Level**, select **High**, **Medium**, or **Low**.
    To customize defense settings, see *Fine-tuning XML defense configuration*, on page 12-16.

11. To mask sensitive XML data, click **Sensitive Data Configuration** and then add namespaces. For details on this task, see *Masking sensitive XML data*, on page 12-19.
12. Click the **Create** button.
    The system adds the XML profile to the security policy.
13. To activate the updated security policy, in the editing context area, click the **Apply Policy** button and then click **OK** to confirm.
14. Next, specify what to associate with the XML profile:
    - **URL**: see *Associating an XML profile with a URL*, on page 12-20, or
    - **Parameter**: see *Associating an XML profile with a parameter*, on page 12-22.

Implementing web services security

Web services security adds another level of protection to XML-based web applications by embedding security-related data within SOAP messages. For web services that Application Security Manager™ protects, you can configure web services security to enable:

- Encryption and decryption of parts of SOAP messages
- Digital signing of parts of SOAP messages
- Verification of parts of SOAP messages using digital signatures

XML digital signatures ensure the integrity of the message data, and can authenticate the identity of the document signer.

You configure web services security on an XML profile in a security policy. Before you configure web services security, you must complete the following tasks:

- Create a security policy with an XML profile: refer to *Configuring security for SOAP web services*, on page 12-3
- Add certificates: refer to *Uploading certificates*, following.
- Configure web services security: refer to *Enabling encryption, decryption, signing, and verification of SOAP messages*, on page 12-7.

◆ Note

Using web services security impacts Application Security Manager system performance.

For details on configuring how to handle web services security errors, refer to *Configuring blocking properties for web services security*, on page 6-45.

Uploading certificates

To use web services security for encryption, decryption, and digital signature signing and verification, you must upload client and server certificates onto the Application Security Manager. The system uses these certificates to process Web Services Security markup in SOAP messages within requests and responses to and from web services.

You must import both client and server certificates to perform encryption and decryption on the Application Security Manager. The certificates you import can be used for any web applications.

To upload certificates

1. In the navigation pane, expand **Application Security**, point to **Options**, then click **Certificates Pool**.
   The Certificates Pool screen opens.
2. Add one server certificate, and a client certificate for each client that you want to access the XML application.

   Note: *The server and client certificates must be* ***.PEM*** *files in x509v3 format. Also, the server certificate should contain the server's private key.*

   For each certificate you want to add, perform these steps:

   a) Click **Add**.
   The Create New Certificate screen opens.

   b) For **Name**, type a name for the certificate.

   c) For **Type**, select **Client** or **Server**.

   d) For the **.PEM File** setting, select **Upload File**, then browse to and upload a certificate, or select **Paste text** to paste a copy of the certificate in the box.

   e) To store the certificate even if it is expired or untrusted, check the **Save Expired/Untrusted Certificate** box.

   f) Click **Add**.

   The system adds the certificate to the certificates pool.

Enabling encryption, decryption, signing, and verification of SOAP messages

You can use web services security features of Application Security Manager to off load encryption and decryption of SOAP messages from the application server. Web services security can also handle verification of digital signatures and digital signing of SOAP messages.

Before you can configure web services security, you must have completed the following tasks:

- Create a security policy with an XML profile, as described in *Configuring security for SOAP web services*, on page 12-3.
- Enable Web Services Security on the XML profile, as described in *Configuring security for SOAP web services*, on page 12-3.
- Upload the required server and client certificates, as described in *Uploading certificates*, on page 12-6.

The task of configuring web services security consists of completing all of the following procedures:

- Beginning web services security configuration
- Configuring web services security credentials
- Configuring web services security requests
- Configuring web services security responses
- Configuring web services security namespaces and elements
- Completing web services security configuration

To begin web services security configuration

1. In the navigation pane, expand **Application Security** and click **XML Profiles**.
2. In the XML Profiles list, click the name of the XML profile for which you want to configure web services security.
   The XML Profile Properties screen opens.
3. Verify that the **Enable Web Services Security** check box is selected.
4. Click **Web Services Security Configuration**.
   The XML Profile Properties screen displays Web Services Security Configuration options.

   Continue to configure credentials.

To configure web services security credentials

On the XML Profile Properties screen, in the Credentials area of the Web Services Security Configuration, configure the certificates the system uses to process SOAP messages in requests and responses.

Tip

*Click the **Certificates Pool** link if you have not yet uploaded certificates.*

1. For **Server Certificate**, select a server certificate from the list.
   The system uses this certificate to decrypt SOAP messages from a web client to a web service, or sign SOAP messages from a web service back to a web client using this certificate.
2. For **Client Certificates**, select names from the **Available** list and then move them into the **Members** list.
   The system uses these certificates to encrypt SOAP messages from a web service to a web client, or verify SOAP messages from a web client to a web service.

   Continue to configure requests.

To configure web services security requests

On the XML Profile Properties screen, in the Request area of the Web Services Security Configuration, configure how you want the system to handle requests.

1. For **Action**, select the action you want the system to perform on SOAP message requests:
   - **Verify and Decrypt** decrypts and verifies digitally signed SOAP messages. We recommend that you use this value.
   - **Decrypt** decodes encrypted SOAP messages.
   - **Verify** validates digitally signed SOAP messages. This option is only available if you imported a server certificate, but no client certificate.
2. For **Role/Actor**, select a role to determine which security headers you want the system to process:
   - **Do not check role/actor**: Process all security headers regardless of the role. This is the default setting.
   - **Custom role/actor**: Process security headers that contain the role you type in the adjacent box.
   - **next**: Process security headers that contain the role **next** or **http://www.w3.org/2003/05/soap-envelope/role/next**.
   - **none**: Process security headers that contain the role **none** or **http://www.w3.org/2003/05/soap-envelope/role/none**.
   - **ultimateReceiver**: Process security headers that contain the role **ultimateReceiver** or **http://www.w3.org/2003/05 /soap-envelope/role/ultimateReceiver**.

3. Check the **Enforce And Verify Defined Elements** box to confirm that elements, defined in the Namespaces and Elements area of the screen and contained in the request, are signed and verified. It also enforces the options **SOAP Body in Request Must Be Signed and Verified** and **Enforce Timestamp In Request**.

To configure web services security responses

On the XML Profile Properties screen, in the Response area of the Web Services Security Configuration, configure how you want the system to handle responses.

1. In the Response area, for **Action**, select the action you want the system to perform on SOAP message responses:
   - **Encrypt** encrypts, in the response, the elements defined in the Namespaces and Elements area of the screen.
   - **Sign** digitally signs, in the response, the elements defined in the Namespaces and Elements area of the screen.
   - **Sign, Then Encrypt** first digitally signs and then encrypts, in the response, the elements defined in the Namespaces and Elements area of the screen. We recommend that you use this value.
   - **Encrypt, Then Sign** first encrypts, then digitally signs, in the response, the elements defined in the Namespaces and Elements area of the screen.

   ***Note**: For the action to occur, you must also check **Apply Action To Defined Elements**.*
2. To limit how long a security header is valid:
   - Check the **Add Timestamp** box.
   - Type the length of time (in seconds) the timestamp should be valid. The default is **300** seconds. If you want the timestamp to be valid for an unlimited amount of time, enter **0**. The maximum value is **100,000,000** seconds.
3. For **Role/Actor**, select a role to insert, in the security header of SOAP messages, information about operations that were performed on the text:
   - **Do not assign role/actor**: Insert, into the existing or created security header, the cryptography (for example, the algorithm, cipher, and keys) that describes the **Action** chosen. This is the default setting.
   - **Assign custom role/actor**: Insert, into the security header, the cryptography (for example, the algorithm, cipher, and keys) that describes the **Action** chosen. In the box, type the value to use for the role/actor attribute in the existing or created security header.
   - **next**: Insert, into the existing or created security header, the cryptography (for example, the algorithm, cipher, and keys) that describes the **Action** chosen. Use **next** for the roll/actor attribute in the security header.

- **none**: Insert, into the existing or created security header, the cryptography (for example, the algorithm, cipher, and keys) that describes the **Action** chosen. Use **none** for the roll/actor attribute in the security header.
- **ultimateReceiver**: Insert, into the existing or created security header, the cryptography (for example, the algorithm, cipher, and keys) that describes the **Action** chosen. Use **ultimateReceiver** for the roll/actor attribute in the security header.

4. For **Signature Algorithm**, select the type of signature algorithm used to sign parts of SOAP messages in responses that match the response elements that you configure in the Namespaces and Elements area of the screen. Select from the following:
   - **RSA-SHA-1** (the default value) uses the RSA public cryptosystem for encryption and authentication with the SHA-1 hash function.
   - **HMAC-SHA-1** uses secret-key hashing with the SHA-1 hash function.

   ***Tip**: Be sure your clients support this type of encryption.*
5. Check the **Apply Action To Defined Elements** box to perform the action you selected in the **Action** setting on responses containing the elements defined in the Namespaces and Elements area of the screen.

   Continue on to configure which namespaces and elements to process in requests and responses.

To configure web services security namespaces and elements

On the XML Profile Properties screen, in the Namespaces and Elements area, configure which parts of the XML document you want the system to process.

1. For **Namespace Mappings**, specify the namespace mappings the system uses for XPath queries:
   a) For **Prefix**, type the namespace prefix.
   b) For **Namespace**, type the URL that the prefix is mapped to.
   c) Click **Add** to add the namespace to the list.
2. Check the **SOAP Body In Request Must Be Signed And Verified** box to verify that the message contains a SOAP body, the SOAP body is digitally signed, and the digital signature is verified. If not, the system issues a **Verification Error** violation.

   ***Note**: For this setting to work, you must also check the **Enforce and Verify Defined Elements** setting in the earlier Request area.*

3. Check **Enforce Timestamp In Request** to check that the SOAP message contains a valid timestamp, the timestamp is not expired, and the digital signature is verified. If the request has no timestamp, the **Missing Timestamp** violation occurs. If the timestamp is expired, the system issues the **Expired Timestamp** violation.

   ***Note**: For this setting to work, you must also check **Enforce and Verify Defined Elements**.*

4. Check the **Apply Action to Entire Response Body Value** box to apply the response action you selected to the whole SOAP message (**/soapenv:Envelope/soapenv:Body**). If not checked, the action occurs only on the elements that are configured on this screen.

5. For the **Elements** setting, perform these steps for each element you want the system to process in responses:

   a) For **Apply to**, select **Response**.

   b) For **XPath**, type an XPath expression to specify which parts of the XML document to encrypt. For details, see *Writing XPath queries*, on page 12-12.

   c) For **Encryption Method**, select whether to encrypt the markup and the text (**With markup**) or the text only (**Value only**).

   d) Click **Add**.

   ***Note**: To process these elements, you must also check **Apply Action To Defined Elements**.*

6. For the **Elements** setting, perform these steps for each element you want the system to process in requests:

   a) For **Apply to**, select **Request**.

   b) For **XPath**, type an XPath expression to specify which parts of the XML document to encrypt. For details, see *Writing XPath queries*, on page 12-12.

   c) Click **Add**.

   ***Note**: To process these elements, you must also check **Enforce and Verify Defined Elements**.*

   Continue on to complete web services security configuration.

To complete web services security configuration

On the XML Profile Properties screen, you can complete web services security configuration.

1. On the XML Profile Properties screen, click the **Update** button. The system updates the XML profile to include the web services security configuration.

2. In the editing context area, click the **Apply Policy** button and confirm to activate the updated security policy.

You have finished configuring web services security on the security policy using the default defense configuration settings. If you want to adjust the settings, refer to *Fine-tuning XML defense configuration*, on page 12-16.

Writing XPath queries

If you want to process specific elements in the XML document, you must write an XPath expression that indicates which parts to process. You specify the XPath in the Web Services Security Configuration area of the XML profile.

When writing XPath queries, you use a subset of the XPath syntax described in the XML Path Language (XPath) standard at **http://www.w3.org/TR/xpath**. Application Security Manager's XPath syntax allows only expressions that correspond to element values.

Here are some rules for writing XPath queries for web services security:

- Express the queries in abbreviated form.
- Map all prefixes to namespaces.
- Use only ASCII characters in queries.
- Write queries to match elements only, not attributes.
- Use wildcards as needed (use ***** for elements and namespaces); for example, **//emp:employee/***.
- Do not use predicates and attributes in queries.

Table 12.1 summarizes the syntax for XPath expressions.

Expression	Description
Nodename	Selects all child nodes of the named node.
/	Indicates XPath step.
//	Selects nodes in the document from the current node that match the selection, no matter where they are.

Table 12.1 *Syntax for XPath expressions*

Table 12.2 shows examples of XPath queries.

Query	Description
/a	Selects the root element **a**.
//b	Selects all **b** elements no matter where they are in the document.
/a/b:*	Selects any element in a namespace bound to prefix **b,** which is a child of the root element **a**.
//a/b:c	Selects elements in the namespace of element **c,** which is bound to prefix **b**, and is a child of element **a**.

Table 12.2 *XPath examples*

Managing SOAP methods

When you upload a WSDL document, the system automatically populates a list of SOAP methods in the validation configuration of the XML profile. Additionally, the system adds the SOAP methods as URLs in the security policy, and automatically associates the XML profile with the URLs.

The system configures into the policy all relevant URLs that it finds in the WSDL and designates them as valid SOAP methods. By default, all methods are enabled, which means that the security policy allows those methods. If you disable a SOAP method, and a request contains that method, the system issues the **SOAP method not allowed** violation, and blocks the request if the enforcement mode is blocking.

Note

Before you can start this task, you must have already uploaded a WSDL document in the XML profile. Refer to ***To create an XML profile for web services security****, on page 12-3, if you have not performed this task.*

To enable or disable a SOAP method

1. In the navigation pane, expand **Application Security** and click **XML Profiles**.
   The XML Profiles screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. Click the name of the profile for which you want to enable or disable one or more SOAP methods.
   The XML Profile Properties screen opens.
4. In the Validation Configuration area, the **Valid SOAP Methods** table lists the SOAP methods used by the WSDL file you uploaded previously. Select or clear the **Enabled** check box for each method that you want to enable (allow) or disable (not allow).

5. Below the Defense Configuration area, click the **Update** button.
   The screen refreshes, and displays the XML Profiles screen.
6. To put the changes into effect immediately, click **Apply Policy** and confirm.
   The system applies the updated security policy.

Configuring security for XML content

You can configure security for applications that use simple XML content rather than web services.

Some XML applications include an XML schema that describes the structure of the XML content. The XML profile can validate whether the incoming traffic complies with that XML schema.

You can upload an existing XML schema file with the following qualifications:

- XML schemas must be in UTF-8 character encoding.
- If the XML schema refers to other XML schemas, you must load the main schema first, then the referenced schema.
- If XML schemas refer to each other, you must upload the main schema twice: first as the main schema, and second as the referenced schema.

To configure security for XML content

1. In the navigation pane, expand **Application Security** and click the create icon (**)** next to **XML Profiles**.
   The Create New XML Profile screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the **Profile Name** box, type a name for the XML profile.
4. For the **Configuration Files** setting, if the application uses an XML schema, perform steps a and b. Otherwise, skip to step 7.

   a) For the **File** option, click **Browse** and navigate to the **.xsd** file.

   Note: *The file you upload must be encoded with UTF-8 character encoding.*

   b) Click **Upload**.
   The screen lists the uploaded file.

   Important*: When an XML schema refers to another XML schema, the system gives you the option of importing it. If circular dependencies exist in the files (for example, schema 1 references schema 2, which contains a reference to back to schema 1) import schema 1, then schema 2, then schema 1 again. This creates a mapping between the files.*

5. If you selected a referenced file type, in the **Import URL** box, type the URL defined in the **schemaLocation** directive.
6. To attempt to locate and use files referenced in the XML schema document, ensure that the **Follow Schema Links** box is checked.

   To use this setting, make sure the DNS server is on the DNS lookup server list, and configure the DNS server on the BIG-IP system (**System>>Configuration>>Device>>DNS)**.

   Tip: *If you disable this setting and the uploaded file refers to other XML schemas, the system lists the referenced files in an error message at the top of the screen.*
7. To permit SOAP messages to contain attachments, check the **Allow Attachments in SOAP Messages** box.
8. Click the **Create** button.
   The system adds the new XML profile to the configuration, and the screen refreshes to display the new profile on the XML Profiles list screen.
9. To put the changes into effect immediately, click **Apply Policy** and then click **OK** to confirm.
   The system applies the updated security policy.

You have finished configuring a security policy for a web application with XML content using the default defense configuration settings. If you want to adjust the settings, refer to *Fine-tuning XML defense configuration*, on page 12-16.

Fine-tuning XML defense configuration

The ***defense configuration*** provides formatting and attack pattern checks for the XML data. The defense configuration complements the validation configuration to provide comprehensive security for XML data and web services applications.

In the defense configuration, the ***defense level*** determines the granularity of the security inspection for the XML application. You can choose **High**, **Medium**, or **Low** and let the system determine the defense level settings. Or you can set the level, then adjust any of the settings to create a **Custom** defense level. The defense level settings, described in Table 12.3, specify the valid properties of the actual XML data or the web services application.

A trade-off occurs between ease of configuration and defense level. The higher the defense level, the more you may need to refine the security policy. For example, if you accept the default defense level of **High**, the XML security is optimal; however, when you initially apply the security policy, the system may generate false-positives for some XML violations.

To fine-tune defense configuration

1. In the navigation pane, expand **Application Security** and click **XML Profiles**.
   The XML Profiles screen opens.
2. In the XML Profiles list, click the name of the XML profile for which you want to modify the advanced defense configuration settings.
   The XML Profile Properties screen opens.
3. From the Defense Configuration list, select **Advanced**.
   The screen refreshes to display additional defense configuration settings.
4. If you want to override specific attack signatures for this XML profile, in the **Global Security Policy Settings** list, select the attack signatures and then click the Move (<<) button to move them into the **Overridden Security Policy Settings** list.

   Tip*: If no attack signatures are listed in the Global Security Policy Settings list, perform an attack signature update, as described in* ***Updating the system-supplied attack signatures****, on page 11-10.*
5. In the **Overridden Security Policy Settings** list, enable or disable each attack signature as needed:
   - **Enabled** specifies that the attack signature is enabled for this XML profile, although it may be disabled in general. The system issues the violation **Attack Signature Detected** when the XML part of a request matches the attack signature.
   - **Disabled** specifies that the attack signature is disabled for this XML profile, although it may be enabled in general.
6. For the **Defense Level** setting, select the protection level you want for the application. The default setting is **High**.

7. Adjust the defense configuration settings as required by your application and traffic. For details, see Table 12.3, on page 12-17.
8. Click the **Update** button.
   The system commits any changes you may have made.
9. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

Table 12.3, describes the defense configuration settings. The Defense Level setting (step 6, in the previous procedure) determines the default values for the settings. A value of **0** in the table indicates unlimited; that is, up to the boundaries of an integer type.

Setting	Description	Default Value: High	Default Value: Medium	Default Value: Low
Defense Level	Specifies the level of protection that the system applies to XML documents, applications, and services. If you change any of the default settings, the system automatically changes the defense level to Custom.	High	Medium	Low
Allow DTDs	Specifies, when enabled, that the XML document can contain Document Type Definitions (DTDs).	Disabled	Enabled	Enabled
Allow External References	Specifies, when enabled, that the XML document is allowed to list external references using operators, such as schemaLocation and SYSTEM.	Disabled	Disabled	Enabled
Tolerate Leading White Space	Specifies, when enabled, that leading white spaces at the beginning of an XML document are acceptable.	Disabled	Disabled	Enabled
Tolerate Close Tag Shorthand	Specifies, when enabled, that the close tag format </>, which is used in the XML encoding for Microsoft® Office Outlook® Web Access, is acceptable.	Disabled	Disabled	Enabled
Tolerate Numeric Names	Specifies, when enabled, that the entity and namespace names can start with an integer (0-9). Note that this is a compatibility option for use with Microsoft® Office Outlook® Web Access.	Disabled	Disabled	Enabled

Table 12.3 *Advanced defense configuration settings*

Setting	Description	Default Value: High	Default Value: Medium	Default Value: Low
Allow Processing Instructions	Specifies, when enabled, that the system allows processing instructions in the XML request. If you upload a WSDL file that references valid SOAP methods, this setting is inactive.	Enabled	Enabled	Enabled
Allow CDATA	Specifies, when enabled, that the system permits the existence of character data (CDATA) sections in the XML document part of a request.	Disabled	Enabled	Enabled
Maximum Document Size	Specifies, in bytes, the largest acceptable document size.	1024000 bytes	10240000 bytes	0 (unlimited)
Maximum Elements	Specifies the maximum number of elements that can be in a single document.	65536	512000	0 (unlimited)
Maximum Name Length	Specifies, in bytes, the maximum acceptable length for element and attribute names.	256 bytes	1024 bytes	0 (unlimited)
Maximum Attribute Value Length	Specifies, in bytes, the maximum acceptable length for attribute values.	1024 bytes	4096 bytes	0 (unlimited)
Maximum Document Depth	Specifies the maximum depth of nested elements.	32	128	0 (unlimited)
Maximum Children Per Element	Specifies the maximum acceptable number of child elements for each parent element.	1024	4096	0 (unlimited)
Maximum Attributes Per Element	Specifies the maximum number of attributes for each element.	16	64	0 (unlimited)
Maximum NS Declarations	Specifies the maximum number of namespace declarations allowed in a single document.	64	256	0 (unlimited)
Maximum Namespace Length	Specifies the largest allowed size for a namespace prefix in the XML part of a request.	256 bytes	1024 bytes	0 (unlimited)

Table 12.3 *Advanced defense configuration settings (Continued)*

Masking sensitive XML data

You can mask sensitive XML data so that it does not appear in the interface or logs. You set this up in the XML profile of any security policy (developed for an XML application).

Note

Before you can start this task, you must have already created an XML profile.

To mask sensitive XML data

1. In the navigation pane, expand **Application Security** and click **XML Profiles**.
   The XML Profiles screen opens.
2. In the XML Profiles list, click the name of the XML profile for which you want to mask sensitive XML data.
   The XML Profile Properties screen opens.
3. Click **Sensitive Data Configuration**.
   The screen displays Sensitive Data Configuration settings.
4. For **Namespace**, select one of the options:
   - **Any Namespace** specifies that sensitive data can appear in any namespace prefix.
   - **No Namespace** specifies that no default namespace prefix has an element or attribute whose value contains sensitive data.
   - **Custom** specifies that the name of the namespace prefix that you type can contain sensitive data.
5. For **Name**,
   a) Select **Element** or **Attribute** to indicate whether the sensitive data appears as a value of an XML element or an attribute.
   b) In the box, type the XML element or attribute whose value contains the sensitive data. Entries in this box are case-sensitive.
6. Click **Add** to add the information you entered in the **Namespace** and **Name** fields to the Sensitive Data table and to the XML profile configuration.
7. Click the **Update** button.
   The system adds the sensitive data information to the XML profile.
8. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

Associating an XML profile with a URL

You can associate XML profiles with explicit URLs and wildcard URLs. The parameter or URL that the XML payload refers to is mostly in the WSDL or the XML schema. When the system receives a request that contains the URL, the system applies the associated XML profile, and generates, if applicable, an XML violation. You can configure the system to verify all requests, or only those requests whose **Content-Type** header contains a configurable string, for example, **text/xml**.

The Security Enforcer applies the XML profile to the entire POST data component in a request. If the **Content-Type** header check fails, the Security Enforcer applies the default HTTP validations for POST data. If you configure the XML profile to validate requests based on the **Content-Type** header values, F5 Networks recommends that you ensure that the security policy also validates POST data.

Tip

You can associate one XML profile with several URLs. You do not need to create a separate XML profile for each URL that you want the system to protect. If you associate an XML profile with a wildcard URL, you can use one XML profile to protect an entire web services application. For more information on wildcard URLs, see ***Configuring wildcard URLs****, on page 9-9.*

To associate an XML profile with a URL

1. In the navigation pane, expand **Application Security** and click **URLs**.
   The Allowed URLs List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Allowed URLs List area, click the name of the URL to which you want to assign an XML profile.
   The Allowed URL Properties screen opens.
4. Check the **Apply XML Profile** box to cause the system to validate XML data in requests to the URL.
   The screen refreshes and provides more settings.
5. For **XML Profile**, select the profile you want to associate with the URL.

   Note*: If you have not created the XML profile, you can click the Create button (+) to create one. The system redirects you back to the URL Properties screen when you are done.*

6. For the **Check XML Content-Type Headers** setting, specify how the system applies the XML profile to requests for this URL.
   - Select **All** if you want the system to inspect all requests.
   - Select **User-defined** and type a string, if you want the system to inspect only those requests whose **Content-Type** header value contains the string you specified. Note that this option has a default setting of ***xml***.
7. Click the **Update** button to save your changes.
8. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

Associating an XML profile with a parameter

You can associate an XML profile with a parameter whose value is XML-encoded. When the system receives a request that contains the parameter, the system applies the XML profile to the parameter value, and if applicable, generates one or more XML violations.

For general information on parameters, refer to Chapter 10, *Working with Parameters*.

To associate an XML profile with a parameter

1. In the navigation pane, expand **Application Security** and click **Parameters**.
   The Parameters List screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the Parameters List area, click the name of the parameter to which you want to assign an XML profile.
   The Parameter Properties screen opens.
4. In the Edit Parameters area, for the **Parameter Value Type**, select **XML value**.
   The screen refreshes, and displays XML profile settings.
5. In the XML Profile area, for the **XML Profile** setting, specify a profile to use; either:
   - Select a profile from the list.
   - Click the Create button (+) to create a new XML profile.

   ***Note**: When you navigate to the Create New XML Profile screen using the Create button (+), the system redirects you to the Parameter Properties screen when you finish creating the new XML profile.*
6. Click the **Update** button to save any changes you may have made.
7. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

Modifying XML security profiles

Web applications change over time, and you may occasionally need to revise or delete an associated XML profile.

Editing an XML profile

You can easily make any necessary changes to the profile, and then apply the updated security policy so that the changes take effect immediately.

Note

Making changes to an XML profile requires external network access to verify the XML schema link. The time needed to complete an XML profile update varies, depending on the size of the WSDL document or XML schema file, and your connection speed.

To edit an XML profile

1. In the navigation pane, expand **Application Security** and click **XML Profiles**.
   The XML Profiles screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the XML Profiles list, in the Profile Name column, click the name of the XML profile that you want to update.
   The XML Profile Properties screen opens.
4. Make any necessary changes to the profile properties, and then click the **Update** button.
   The system saves any changes you may have made.
5. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

Deleting an XML profile

If you no longer need a specific XML profile, you can remove it entirely from the configuration. F5 Networks recommends that before you delete an XML profile, you remove the profile from any URLs or parameters with which the profile is associated.

To delete an XML profile

1. On the navigation pane, click **XML Profiles**.
   The XML Profiles screen opens.
2. In the editing context area, verify that the edited security policy is the one you want to update.
3. In the XML Profiles area, in the Select column (far left), check the box next to the profile that you want to remove, and then click the **Delete** button.
   The system displays a popup confirmation screen.
4. To put the changes into effect immediately, click **Apply Policy** then click **OK** to confirm.
   The system applies the updated security policy.

13

Refining the Security Policy Using Learning

- Overview of the learning process
- Working with learning suggestions
- Accepting or clearing learning suggestions
- Working with entities in staging or with tightening enabled
- Processing learning suggestions that require user interpretation
- Viewing ignored entities
- Adding and deleting ignored IP addresses

Overview of the learning process

You can use learning process resources to help while building a security policy. When you send client traffic through the Application Security Manager™, the Learning data provides information on requests or responses that do not comply with the current security policy and triggered a violation. The reason for triggering a violation can be either a false positive (typically seen during the process of building a policy), or an actual attack on the site.

The system generates learning suggestions for requests that cause violations and do not pass the security policy checks. You examine the requests that cause learning suggestions, and then use the suggestions to refine the security policy. In some cases, learning suggestions may contain recommendations to relax the security policy due to attacks. When dealing with learning suggestions, make sure to relax the policy only where false positives occurred, and not in cases where a real attack caused a violation.

The learning process includes the resources described in Table 13.1.

Resource	Description
Learning Manager	An internal system process that examines the security policy violations that the system identifies, and generates learning suggestions based on those policy violations. As visitors move through the web application, the Learning Manager captures requests that contravene the current security policy settings, and records the learning suggestions on the Traffic Learning screen.
Traffic Learning screen	A screen that displays learning suggestions that the Learning Manager generates. The learning suggestions are categorized by violation type, and can represent actual threats or false-positives. Learning suggestions are for the currently active security policy. When you accept a learning suggestion, you are updating the currently active security policy.
Staging-Tightening screen	A screen that summarizes the security policy entities in staging or with tightening enabled, that may have learning suggestions, and may be ready to be enforced. For file types, parameters, URLs, and cookies, you can review the entities, and decide whether to add them to the security policy.
Ignored Entities screen	A screen that lists the file types, URLs, and flows that you have instructed the Learning Manager to disregard, that is, to stop generating learning suggestions for. Typically, the ignored entities are items that you do not want to be a part of the security policy.
Ignored IP Addresses screen	A screen that lists IP addresses that you have instructed the system to ignore. The system does not generate learning suggestions for traffic sent from these IP addresses.
View Full Request Information screen	A screen that lists any violations and details associated with a request. You can review this information, and then if you want to accept the learning suggestion, click the Learn button to update the active security policy. To display the View Full Request Information screen, from the Reporting Requests screen, click a Requested URL in the Requests List.

Table 13.1 *Learning process resources*

Working with learning suggestions

The Learning Manager generates learning suggestions when the Learn flag is enabled for the violations on the Policy Blocking Settings screen. (See *Configuring the blocking actions*, on page 6-43, for how to set the flag.) When the system receives a request that triggers a violation, the Learning Manager then updates the Traffic Learning screen with learning suggestions based on the violating request information (see Figure 13.1 for an example screen). From this screen, you can review the learning suggestions to determine whether the request triggered a legitimate security policy violation, or if the violation represents a need to update the security policy.

Making decisions about which learning suggestions to use requires some general understanding of application security, and specific knowledge of the protected application (for example, recognizing valid traffic). Often, you should consider accepting a learning suggestion when you see that it has occurred multiple times, from many different source IP addresses. Repeated learning suggestions typically indicate valid traffic behavior that warrants relaxing the security policy.

The Traffic Learning screen also displays violations for which the system does not generate learning suggestions. Typically, these violations are related to RFC compliance and system resources, rather than to security policy entities. The system displays these violations along with the learning suggestions to ease the security policy management tasks.

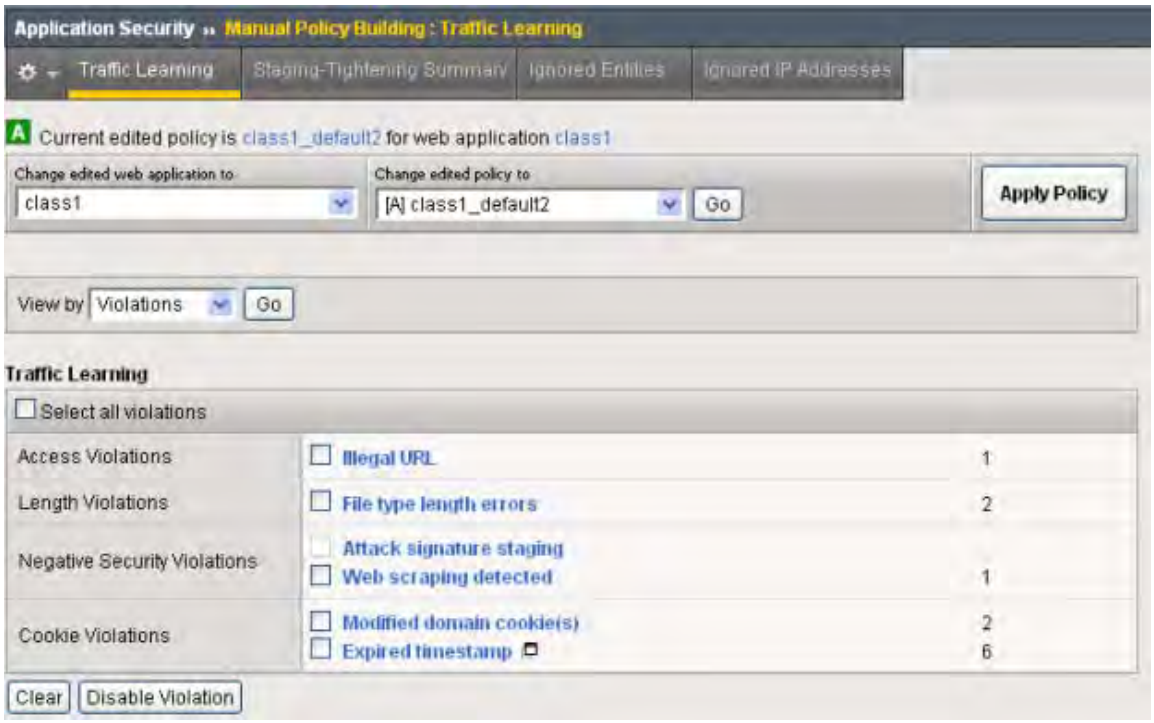


Figure 13.1 *Example of Traffic Learning screen*

Note

The Traffic Learning screen displays violations only when the system has detected them in a request.

To view learning suggestions

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to review the learning suggestions.
3. In the **View by** box, select how you want the system to display the triggered violations.
4. In the Traffic Learning area, click a violation hyperlink to view the specific elements in the request that triggered the security policy violation and the corresponding learning suggestion.
   The system displays the learning suggestion details (for learnable violations), or a list of requests (for unlearnable violations).

Note

*In learning suggestions and on the View Full Request Information screen, the Application Security Manager displays and processes non-printable characters, that is, control characters, in the same manner as it displays and processes other characters. For example, the system displays the space character as **0x20**.*

Viewing all requests that trigger a specific learning suggestion

Before you process a learning suggestion, it is very helpful to examine the details of the request that caused the learning suggestion. For learnable violations, you can review all of the requests that trigger a specific learning suggestion by clicking the number of occurrences of that learning suggestion. For unlearnable violations, click the violation itself to review the requests.

To view all of the requests that triggered a specific learning suggestion

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to review the learning suggestions.
3. In the Traffic Learning area, click a violation hyperlink to view either the Requests List for unlearnable violations, or the specific elements in the request that triggered the security policy violation and the corresponding learning suggestion.
4. In the Occurrences column, click the number.
   The Requests List popup screen opens, and displays all of the requests that triggered the learning suggestion.

Viewing the details of a specific request

On the View Full Request Information screen, you can view some or all of the following information:

- Triggered violations, and their respective blocking actions
- Full request contents
- Requested URL
- Web application name
- Support ID number
- Source and destination IP addresses
- Country
- Time
- Flags
- Severity
- Response status code
- Potential attacks
- Any parameter and value pairs, their triggered violations, and their parameter levels
- Attack signatures, if applicable
- XML data, if applicable

To view a request that triggered a learning suggestion

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to review the learning suggestions.
3. In the Traffic Learning section, click a violation hyperlink to view either the request or the specific elements in the request that triggered the security policy violation and the corresponding learning suggestion.
   The screen refreshes, and the system displays the request or request elements that caused the learning suggestions.
4. In the Occurrences column, if available, click the number.
   The Requests List popup screen opens, and displays all of the requests that contained an item that triggered the learning suggestion.

   ***Note**: Some violations have no Occurrences number.*
5. In the Requests List area of the popup screen, in the URL column, click a URL link.
   The View Full Request Information screen opens in the popup screen, where you can review the request that triggered the learning suggestion.

   Figure 13.2 shows an example of the View Full Request Information popup screen. It shows the violations associated with the request, and details about the request.
6. For each violation with a **Learn** button, click **Learn** to open the violation learning screen where you can accept or clear the learning suggestions for the security policy one value at a time.
7. To view the actual contents of the request, click **Full Request**.
8. If you are sure that the request is trusted, click **Accept**.
   The system then directs you to the Automatic Policy Building Status screen where you can see the status of the Policy Builder.

   ***Tip**: The system does not display the **Accept** button when the Policy Builder is already running or if the request is legal.*
9. To remove Learning suggestions without changing the security policy, select the ones to remove, and then click the **Clear** button.

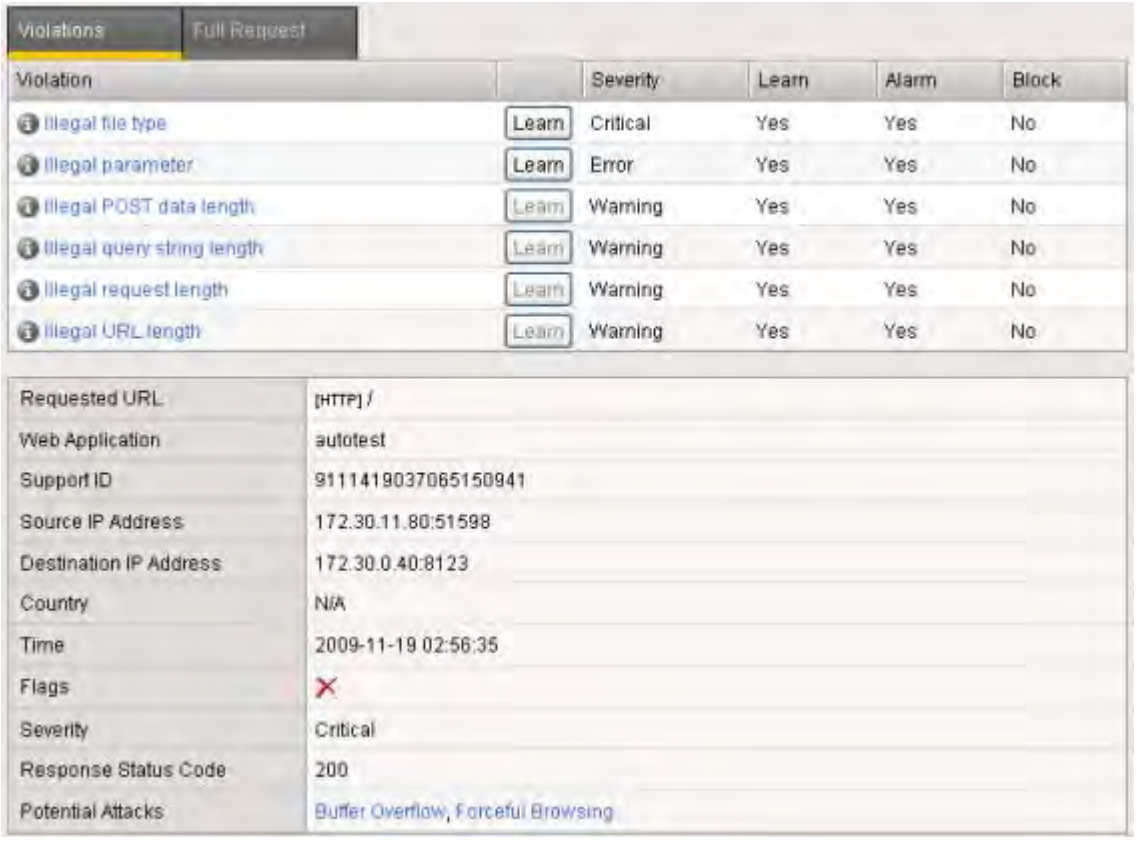


Figure 13.2 *Example of the View Full Request Information screen*

Viewing all requests for a specific web application

If you want to review all of the requests for a web application that trigger learning suggestions, you can do so on the Requests screen.

To view all requests for a specific web application

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens.
2. In the editing context area, ensure that the web application and security policy are those for which you want to review requests.
3. In the **Filter** list, select **Custom**.
4. For the **Web Applications** setting, select the name of the web application for which you want to see requests.

5. Click the **Go** button.
   The screen refreshes, and in the Requests List area, you see the requests for the selected web application only.

Tip

*For more information about working with the Requests screen, and general reporting tools, refer to Chapter 15, **Displaying Reports**.*

Accepting or clearing learning suggestions

The Learning Manager generates learning suggestions throughout the life of the security policy. When the system detects violations of a security policy, the violations may be related to a real attack, and may therefore warrant more careful inspection before being accepted into the security policy.

You can review learning suggestions (violations) on the Traffic Learning screen, and accept or clear each suggestion, as described following. You can also view learning suggestions from the Staging-Tightening Summary screen, as described in *Working with entities in staging or with tightening enabled*, on page 13-9.

Accepting a learning suggestion

By default, learning suggestions are presented for the active policy. When you accept a learning suggestion, the system updates the current edited security policy to accept the request entity that triggered the violation. It is possible to accept learning suggestions for a policy that is not active, however, so care must be taken to select the policy for which you want to accept learning suggestions.

Note

*Some violations do not provide learning suggestions because you must manually review the requests that caused them. See **Processing learning suggestions that require user interpretation**, on page 13-15, for more information.*

To accept learning suggestions

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.

3. Click a violation hyperlink.
   The learning suggestions properties screen opens. Note that the screens vary depending on the violation.
4. Select one or more learning suggestions, and then click the **Accept** or **Allow** button.
   The system updates the security policy with the element in the request that caused the learning suggestion.

   ***Tip**: Some learning suggestion screens include an **Accept All** button that you can click to accept all of the suggestions on the screen.*

Clearing a learning suggestion

When you clear a learning suggestion, the system deletes the learning suggestion, and does not update the security policy. The Learning Manager continues to generate learning suggestions for future instances of the violation.

To clear learning suggestions

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.
3. Click a violation hyperlink.
   The violation properties screen opens.
4. Select one or more learning suggestions, and then click **Clear**.
   A Confirm Delete popup screen opens.

   ***Tip**: Some learning suggestion screens include a **Clear All** button that you can click to clear all of the suggestions on the screen.*
5. Click **OK**.
   The system deletes the learning suggestion.

◆ Note

*For a description of the violation types, refer to Appendix A, **Security Policy Violations**.*

Working with entities in staging or with tightening enabled

You use the Staging-Tightening summary (shown in Figure 13.3) to review file types, URLs, parameters, and cookies that are in staging or with tightening enabled, and you can delve into the details to see if you want to add these entities to the security policy. You can add selected entities to the security policy, or you can enforce all of the entities that are ready to be enforced.

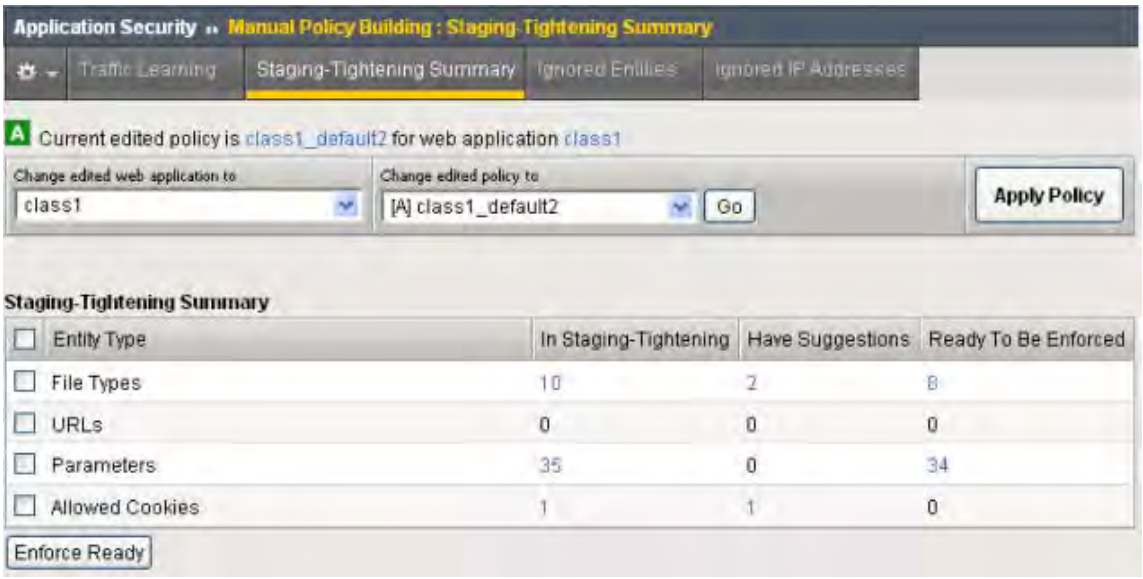


Figure 13.3 *Example Staging-Tightening Summary*

You can click the numbers in the columns to display details about the entities that are in staging or with tightening enabled. For example, Figure 13.4 shows the learning suggestions that are displayed when you click the number link in the Have Suggestions column of the file types entity.

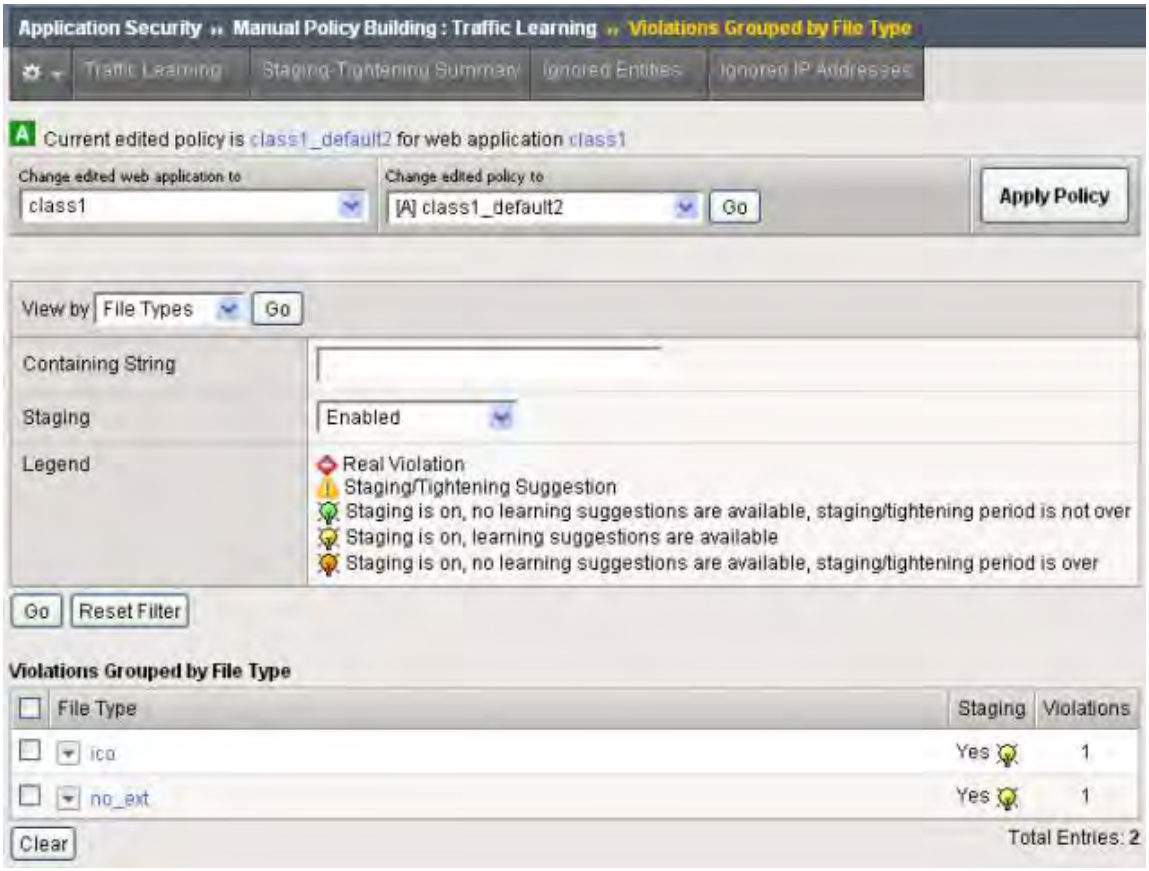


Figure 13.4 *File type learning suggestions*

When you look at the learning suggestions, you can clear them or go back to the staging-tightening summary and enforce the entities. You can also click a learning suggestion in the list to have the security policy learn it, as described in *Accepting a learning suggestion*, on page 13-7.

Understanding tightening

You can perform tightening on wildcard entities (file types, URLs, parameters, and cookies) to learn explicit entities. When you enable tightening for a wildcard entity, and the system receives a request that contains an entity that matches the wildcard entity, the system generates a learning suggestion for the found entity. You can then review the new entities, and decide which are legitimate entities for the web application.

Tightening allows you to develop a more specific policy that is more accurate and in alignment with the traffic. Such a policy can provide better security, but requires more tuning to make sure all the specific entities that you add are accurately configured.

If the Policy Builder is active, and the traffic source is trusted (either by definition or because of heuristic decisions), the Policy Builder automatically adds the new specific entity to the security policy.

Each security policy can have wildcards for file types, URLs, parameters, and cookies. When you create a security policy using the Deployment wizard, the system enables tightening on wildcard entities (depending on the scenario you select). As traffic is sent to the web application, the system learns the explicit properties of the file types, URLs, parameters, and cookies.

◆ Tip

Use tightening on wildcard entities to build the security policy with explicit entities of this type. For additional information on wildcard entities, see Chapter 9, ***Working with Wildcard Entities****.*

Understanding staging

You can perform staging on file types, URLs, and parameters to learn properties of entities, such as:

- For file types, learn file type lengths (URL length, request length, query string length, or POST data length).
- For URLs, learn meta characters (wildcard URLs only).
- For parameters, learn parameter settings.

When an entity is in staging, the system does not block any requests for this entity. Instead, it posts learning suggestions for staged entities on the Learning screens.

◆ Tip

Use staging on wildcard entities to build the security policy without specifying explicit entities of this type.

Staging is also useful when a site update occurs for a web application. Without staging, you might have to change the blocking policy enforcement mode to transparent for the entire web site to discover any new URLs or parameters in the updated web application. With staging, you can add any new URLs or parameters to the security policy, and place only the new entities in staging allowing the system to generate learning alerts.

Reviewing staging and tightening status

If a file type, URL, parameter, or cookie is in staging or has tightening enabled, the system displays a light bulb icon in the Staging or Tightening column of the file types, URLs, parameters, or cookies. For example, Figure 13.5 shows the Allowed File Types List with three files types in staging.

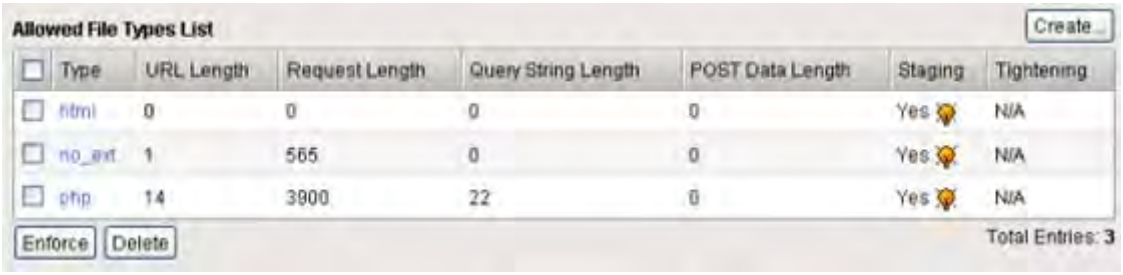


Figure 13.5 *Allowed file types with staging enabled*

The color of the light bulb provides details about the status of the file type, URL, or parameter.

- Green indicates that no learning suggestions are available, and the staging period is not over.
- Yellow indicates that learning suggestions are available. Move the cursor over the light bulb icon to see whether the staging period is over, or not.
- Orange indicates that no learning suggestions are available and the staging period is over. This entity is ready to be taken out of staging, and be enforced.

Move the cursor over the light bulb to see when the entity was placed in staging and the last time the properties of this entity were changed (the Last staging event time date and time). Figure 13.6 shows an example of the information that you can view.

Figure 13.6 *Staging status information*

Adding new entities to the security policy from staging or tightening

After you create a security policy and traffic is sent to the web application, you can review the entities that are in staging or with tightening enabled and add the entities to the security policy. When the staging or tightening period is over and no learning suggestions have been added for seven days, the file type, URL, parameter, or cookie is considered ready to be enforced. You can enforce the entities one at a time or, if they are ready to be enforced, you can enforce them all at once.

To enforce selected file types, URLs, parameters, or cookies in staging or with tightening enabled

1. In the navigation pane, expand **Application Security**, point to **Manual Policy Building** and click **Staging-Tightening Summary**.
   The Staging-Tightening Summary screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.
3. In the Staging-Tightening Summary, check to see if a number appears in the **In Staging-Tightening** column.
   A number greater than zero indicates that entities of that type are in staging or with tightening enabled.
4. Click the number in the **In Staging-Tightening** column.
   The allowed file types, URLs, parameters, or cookies list opens showing the entities that you can enforce.
5. Select the entities to change in the security policy.
6. Click **Enforce**.
   The system takes the following actions:
   - Removes from staging all entities (explicit and wildcard) whose staging period is over.
   - Deletes from the security policy wildcard entities with tightening enabled.

To enforce all file types, URLs, parameters, and cookies that are ready to be enforced

1. In the navigation pane, expand **Application Security**, point to **Manual Policy Building** and click **Staging-Tightening Summary**. The Staging-Tightening Summary screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.
3. Select the entity types to change in the security policy.
4. Click the **Enforce Ready** button.
   The system takes the following actions:
   - Removes all entities whose staging period is over.
   - Deletes wildcard entities with tightening enabled from the security policy.

Processing learning suggestions that require user interpretation

For a few violations, the learning suggestions do not represent an update to the security policy. Instead, the violations require user interpretation. The Policy Builder cannot resolve these violations.

The violations that require user interpretation are:

- RFC Violations
  - Cookie not RFC-compliant
  - Mandatory HTTP header is missing
- Access Violations
  - CSRF attack detected
  - CSRF authentication expired
  - Illegal entry point
  - Illegal flow to URL
  - Illegal HTTP status in response
  - Illegal session ID in URL
  - Login URL bypassed
  - Login URL expired
- Length Violations
  - Illegal cookie length
  - Illegal header length
- Input Violations
  - Failed to convert character
  - Illegal attachment in SOAP message
  - Illegal dynamic parameter value
  - Illegal empty parameter value
  - Illegal meta character in header
  - Illegal number of mandatory parameters
  - Illegal parameter data type
  - Illegal parameter numeric value
  - Illegal query string or POST data
  - Illegal repeated parameter name
  - Illegal static parameter name
  - Malformed XML data
  - Maximum login attempts are exceeded
  - Null in multi-part parameter value
  - Parameter value does not comply with regular expression
  - SOAP method not allowed

  - Web scraping detected
  - Web Services Security failure
  - XML data does not comply with format settings
  - XML data does not comply with schema or WSDL document
- Cookie Violations
  - ASM Cookie Hijacking
  - Expired timestamp
  - Modified ASM cookie
- Negative security violations
  - Information leakage detected
  - Virus detected

For these violations, F5 Networks recommends that you review the violations, and determine whether they represent legitimate violations or false-positives. You can disable these violations if they are not applicable to your web application, which turns off the blocking policy so that you are no longer notified of requests that trigger the violation. Alternately, you can clear the learning suggestions, and Application Security Manager continues to issue learning suggestions for the requests.

Note

Application Security Manager does not generate learning suggestions for requests if the web server sends an HTTP response that includes status codes in the 4xx-5xx range.

Disabling violations

If you do not want the system to display the violations that require user interpretation, you can disable the violation. The **Disable Violation** button disables all flags on the selected violation. The system then ignores future instances of the violation, and passes the requests on to the web application resources.

WARNING

Disabling violations or signature sets can have severe consequences. Be sure that you understand the ramifications of the disabling action before completing it.

Tip

The Traffic Learning screen displays learning suggestions only if the traffic has triggered a violation.

To disable a violation

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.
3. In the Traffic Learning area, check the box next to the violation name that you want to disable.
4. Click the **Disable Violation** button.
   A confirmation popup screen opens.
5. Click **OK**.
   The screen refreshes, and you no longer see the violation in the Traffic Learning area.

   ***Tip**: You can navigate to the **Policy**>>**Blocking**>>**Settings** screen to see that all flags on the selected violation are unchecked.*
6. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.
   A confirmation popup screen opens.
7. Click **OK**.
   The system applies the updated security policy.

Clearing violations

When you clear a violation, the system deletes the violation, but does not update the security policy. The Security Enforcer continues to generate alarms for future instances of the violation, and the Learning Manager continues to generate learning suggestions relative to the violation.

To clear a violation

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one you want to update.
3. In the **View by** list, select whether to view by **Violations, Parameters**, **URLs**, or **File Types**.
4. In the violations list, check the box next to a violation, and then click **Clear**.
   A Confirm Delete popup screen opens.
5. Click **OK**.
   The system deletes the learning suggestion.

Viewing ignored entities

When you reject a learning suggestion for a URL, a file type, or a flow, the Application Security Manager adds the rejected item to the ignored entities list. When the system receives subsequent requests for those rejected items, the system no longer generates learning suggestions related to the rejected items. The system does, however, continue to log the requests.

◆ Note

Items in the Ignored Entities list are ignored for the entire web application, including all of the security policies associated with it.

To view ignored entities

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to review ignored entities.
3. On the menu bar, click **Ignored Entities**.
   The Ignored Entities screen opens showing the number of ignored entities for file types, URLs, and parameters. If ignored entities exist for an entity type, that type becomes a link that you can click to view a list of all entities logged within that category.

Removing items from the ignored entities list

If you want the system to start generating learning suggestions for items that were previously added to the ignored entities list, you can remove those items from the list.

To remove an item from the ignored entities list

1. In the navigation pane, expand **Application Security** and click **Manual Policy Building**.
   The Traffic Learning screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to review ignored entities.
3. On the menu bar, click **Ignored Entities**.
   The Ignored Entities screen opens.
4. For the entity type whose ignored entities you want to remove, check the Select box (far left), and click the **Delete** button.

Adding and deleting ignored IP addresses

For each web application, you can set up a list of IP addresses that you want the system to ignore after enforcement. The system does not generate learning suggestions for traffic sent from these IP addresses.

Creating a list of ignored IP addresses is useful, for example, if your company performs penetration testing using manual or automatic scanners. When you add the IP address of the scanner, the system no longer generates learning suggestions for traffic from the scanner, but does make learning suggestions for other legitimate production traffic.

To add a list of IP addresses to ignore during learning

1. In the navigation pane, expand **Application Security,** point to **Manual Policy Building**, then click **Ignored IP Addresses**.
   The Ignored IP Addresses screen opens.
2. In the editing context area, ensure that the current edited web application is the one for which you want to add IP addresses to ignore.
3. Click the **Create** button.
   The New Ignored IP Address screen opens.
4. Type the IP address that you want the system to ignore, and click **Create**.
   The system adds the IP address to the list of ignored IP addresses.

To delete ignored IP addresses

1. In the navigation pane, expand **Application Security,** point to **Manual Policy Building**, then click **Ignored IP Addresses**.
   The Ignored IP Addresses screen opens where you can see a list of IP addresses that the system ignores during the learning process.
2. In the editing context area, ensure that the current edited web application is the one for which you want to delete an ignored IP address.
3. Check the Select box (far left) for the IP addresses you want to remove, and click the **Delete** button.
   After you confirm, the IP address is removed from the list.

14

Configuring General System Options

- Overview of general system options
- Configuring interface and system preferences
- Configuring external anti-virus protection
- Configuring user accounts for security policy editing
- Configuring logging profiles for web application data
- Setting event severity levels for security policy violations
- Viewing the application security logs
- Validating regular expressions
- Configuring an SMTP mail server

Overview of general system options

The Application Security Manager™ includes general system options that apply to the overall application security configuration. You can perform the following tasks to configure general system options:

- Configure the default Application Security Manager user interface and system preferences. See *Configuring interface and system preferences*, on page 14-2, for more information.
- Configure the Application Security Manager to connect with an Internet Content Adaptation Protocol (ICAP) server to check requests for viruses. See *Configuring external anti-virus protection*, on page 14-3, for more information.
- Create a user account that permits only security policy editing. See *Configuring user accounts for security policy editing*, on page 14-4, for more information.
- Create custom logging profiles for requests data. See *Configuring logging profiles for web application data*, on page 14-5, for more information.
- Set the severity levels of log entries for security policy violations. See *Setting event severity levels for security policy violations*, on page 14-11, for more information.
- Review the system logs for application security events and user activity. See *Viewing the application security logs*, on page 14-12, for more information.
- Use the RegExp Validator to verify that your regular expression syntax is accurate. See *Validating regular expressions*, on page 14-13, for more information.
- Enable the SMTP mailer and configure an SMTP mail server. See *Configuring an SMTP mail server*, on page 14-14, for more information.

Some of the overall system configuration tasks are described in other chapters, because they relate to other tasks described there. You can perform the following additional general configuration tasks:

- Upload client and server certificates for web services security. See *Uploading certificates*, on page 12-6, for more information.
- Update and organize attack signatures. See Chapter 11, *Working with Attack Signatures*, for more information.
- Review the system's internal parameters. See Appendix D, *Internal Parameters for Advanced Configuration*, for more information.

Configuring interface and system preferences

You can change the default user interface and system preferences for the Application Security Manager.

To configure interface and system preferences

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Preferences**.
   The Preferences screen opens.
2. For **Records Per Screen**, type the number of entries to display (**1-100**). The default value is **20**.
   This setting affects the maximum number of web applications, file types, URLs, parameters, flows, headers, and XML profiles to display in lists throughout the Application Security Manager.
3. For **Records Per Requests Screen**, type the number of requests to display (**1-1000**). The default value is **500**.
   This setting affects the maximum number of requests that appear in the Requests List (**Reporting>>Requests**).
4. For **Titles Tooltip Settings**, select one of the options for how to display tooltips:
   - **Show tooltip icons**: Display an icon if a tooltip is available for a setting, and show the tooltip when you move the cursor over the icon. This is the default setting.
   - **Show tooltips on title mouseover**: Display a tooltip when you move the cursor over a setting on the screen.
   - **Do not show tooltips**: Never display tooltips or icons.
5. For **Advanced by Default**, select whether to display all possible settings (**Advanced**) or the **Basic** settings on screens with that option.
6. If the BIG-IP system is in a redundant configuration and you want to display a message telling you to synchronize the two systems when a security policy was updated but not applied, check the **Recommend Sync When Policy Not Applied** box.
7. Specify what information to record in the Syslog (**/var/log/asm**):
   - To log system data and configuration changes made to all security policies, check the **Write all changes to Syslog** box.
   - To log system data only, clear the **Write all changes to Syslog** check box. This is the default setting.
8. Click **Save** to keep your changes.

Configuring external anti-virus protection

You can configure the Application Security Manager to connect with an Internet Content Adaptation Protocol (ICAP) server to check requests for viruses. If the **Virus Detected** violation is enabled for that web application's security policy, the system sends requests with file uploads to an external ICAP server for inspection. The ICAP server examines the requests for viruses and, if the ICAP server detects a virus, it notifies the Application Security Manager, which then issues the **Virus Detected** violation.

The default ICAP server is configured for McAfee anti-virus protection. If the ICAP server supports anti-virus protection using different software, you must change the value of the **virus_header_name** internal parameter. Refer to Appendix D, *Internal Parameters for Advanced Configuration*, for information about internal parameters.

◆ Note

Anti-virus protection may slow down file transfers because the ICAP server examines all requests with file uploads.

To configure external anti-virus protection

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Anti-Virus Protection**.
   The Anti-Virus Protection screen opens.
2. Configure either the host name or the IP address of the ICAP server:
   - For **Server Host Name**, type the ICAP server host name in the format of a fully qualified domain name.

     Note: *If using the host name only, you must also configure a DNS server on the BIG-IP system. Expand* **System**, *point to* **Configuration**, **Device**, *then click* ***DNS***. *If DNS is not configured, you must include the IP address.*
   - For **Server IP Address**, type the IP address of the ICAP server.
3. For **Server Port Number**, type the port number of the ICAP server.
4. If you want to perform virus checking even if it may slow down the web application, check the **Guarantee Enforcement** box.
5. Click **Save** to save the ICAP server configuration.
6. In the navigation pane, point to **Policy**, and then click **Blocking**.
   The Blocking Settings screen opens.
7. For each web application and policy that you want to have anti-virus protection, complete these tasks:

   a) In the editing context area, ensure that the edited web application and security policy are the ones for which you want anti-virus protection.

b) For the **Virus Detected** violation (near the bottom of the screen), enable either or both of the Alarm and Block check boxes. For details on setting up blocking, refer to *Configuring the blocking policy*, on page 6-41.

c) Click **Save** to save the blocking policy.

d) To put the anti-virus protection into effect immediately, click the **Apply Policy** button in the editing context area.

Configuring user accounts for security policy editing

User accounts on the BIG-IP system are assigned a user role that specifies the authorization level for that account. While an account with the user role of Administrator can access and configure everything, you may want to further specialize administrative accounts. You must have Administrator access to create accounts on the BIG-IP system.

The BIG-IP system provides two user roles specifically for security policy management:

- **Web Application Security Editor**
  Grants users permission to view and configure most parts of the Application Security Manager, on specified partitions.
- **Web Application Security Administrator**
  Grants users permission to view and configure all parts of the Application Security Manager, on all partitions. With respect to application security objects, this role is equivalent to the Administrator role.

For additional information on user roles and user management, refer to the ***TMOS® Management Guide for BIG-IP® Systems***.

To configure user accounts for security policy editing

1. In the navigation pane, expand **System**, and then click **Users**.
   The User List screen opens.
2. Click the **Create** button.
   The New User screen opens.
3. For the **User Name** setting, type the name for the account.
4. For the **Password** setting, type and confirm the account password.
5. For the **Role** setting, select the appropriate role:
   - To limit security policy editing to the current partition, select **Web Application Security Editor**.
   - To allow security policy editing on all partitions, select **Web Application Security Administrator**.

6. If you selected **Web Application Security Editor**, then in **Partition Access**, select the partition in which to allow the account to create security policies.
7. Click **Finished**.
   The User List screen opens and includes the new user account in the list.

Configuring logging profiles for web application data

Logging profiles specify how and where the system stores request and violation data for web applications. When you configure a web application, you select the logging profile for that web application. You can use one of the system-supplied logging profiles, or you can create a custom logging profile. Note that the system-supplied logging profiles log data locally. For more information on selecting the logging profile for a web application, refer to *Specifying the logging profile for a web application*, on page 4-4.

Additionally, you can choose to log the request data locally, on a remote storage system (such as a syslog server), on a reporting server (as key/value pairs), or on an ArcSight server (in CEF format).

A logging profile has two parts: the storage configuration and the storage filter. The storage configuration specifies where the logs are stored, either locally or remotely. The storage filter determines what information gets stored.

Creating a logging profile for local storage

You can create a logging profile to store request data on the local BIG-IP system. When you store the request data locally, the logging utility may compete for system resources. You can use the **Guarantee Logging** setting to ensure that the system logs the requests in this situation.

◆ Note

*Enabling the **Guarantee Logging** setting may cause a performance reduction if you have a high traffic-volume application.*

To view logs stored locally, refer to *Viewing the application security logs*, on page 14-12.

To create a logging profile for local storage

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Logging Profiles**.
   The Logging Profiles screen opens.
2. Above the Logging Profiles area, click the **Create** button.
   The Create New Logging Profile screen opens.

3. For the **Configuration** setting, select **Advanced**.
4. In the Configuration area, for the **Profile Name** setting, type a unique name for the logging profile.
5. To ensure that the system logs requests for the web application, even when the logging utility is competing for system resources, check the **Guarantee Logging** box.

   ***Note**: Enabling this setting may slow access to the associated web application.*
6. In the Storage Filter area, make any changes as required. (See *Configuring the storage filter*, on page 14-10, for details.)
7. Click the **Create** button.
   The screen refreshes, and displays the new logging profile on the Logging Profiles screen.

Configuring a logging profile for remote storage

You can create a logging profile to store information remotely on syslog servers in Comma Separated Value (CSV) format or a user-defined format. When you configure a logging profile for remote storage, the system stores request data for the associated web application on a separate remote management system, where you can view the files.

◆ Note

The logging profile for remote storage relies on external systems to perform the actual logging. The configuration and maintenance of the external logging servers is not the responsibility of F5 Networks.

To configure a logging profile for remote storage

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Logging Profiles**.
   The Logging Profiles screen opens.
2. Above the Logging Profiles area, click the **Create** button.
   The Create New Logging Profile screen opens.
3. For the **Configuration** setting, select **Advanced**.
4. For the **Profile Name** setting, type a unique name for the logging profile.
5. Check the **Remote Storage** box, and make sure the **Type** is set to **Remote**.
   The screen displays additional settings.
6. If you do not want data logged locally as well as remotely, clear the **Local Storage** check box.
7. For the **Protocol** setting, select the protocol that the remote storage server uses: **TCP** (the default setting), **UDP**, or **TCP-RFC3195**.

8. For the **Server IP** setting, type the IP address of the remote storage server.
9. For the **Server Port** setting, type a port number or use the default value, **514**.
10. For the **Facility** setting, select the **syslog** facility where you want to store the logged traffic. The possible values are **LOG_LOCAL0** through **LOG_LOCAL7**.

    ***Tip**: If you have more than one web application, and you configure remote logging for both applications, you can use the facility filter to sort the data for each.*
11. For the **Storage Format** setting, from the **Available Items** list, select the data items to include in the log. Use the Move button (<<) to add the data items to the **Selected Items** list.

    Optionally, specify the log format for the data items, by selecting one of the following options:

    - **Predefined**: If you select this option, specify the delimiter to separate the data items in the log (the default delimiter is comma). You may not use the % character. This is the default value.
    - **User-defined**: If you select this option, in the **Selected Items** box, type any text you want to appear between the items, with surrounding percent (**%**) characters (for example,**%Request%**).
12. To ensure that the system logs requests for the web application, even when the logging utility is competing for system resources, check the **Guarantee Logging** box.

    ***Note**: Enabling this setting may slow access to the associated web application.*
13. Optionally, adjust the maximum request, header, and query string sizes, and maximum entry length settings. (Refer to online help for details on the settings.)
14. If you want the system to log details (including the start and end time, number of dropped requests, attacking IP addresses, and so on) about brute force attacks, DoS attacks, IP enforcer attacks, or web scraping attacks, check the **Report Detected Anomalies** box.
15. In the Storage Filter area, make any changes as required. (See *Configuring the storage filter*, on page 14-10, for details.)
16. Click the **Create** button.
    The screen refreshes, and displays the new logging profile on the Logging Profiles screen.

Configuring a logging profile for a reporting server

If your network uses a third party reporting server (for example, Splunk), you can configure a logging profile to store the log information on the reporting server using the key-value pair storage format.

Note

This logging profile relies on external reporting server to perform the actual logging. The configuration and maintenance of the reporting server is not the responsibility of F5 Networks.

To create a logging profile for a reporting server

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Logging Profiles**.
   The Logging Profiles screen opens.
2. Above the Logging Profiles area, click the **Create** button.
   The Create New Logging Profile screen opens.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes to display additional settings.
4. For the **Profile Name** setting, type a unique name for the logging profile.
5. Check the **Remote Storage** box, and for the **Type** setting, select **Reporting Server**.
   The screen displays additional settings.
6. If you do not want data logged locally as well as remotely, click to clear the **Local Storage** check box.
7. For the **Protocol** setting, select the protocol that the reporting server uses: **TCP** (the default setting), **UDP**, or **TCP-RFC3195**.
8. For the **Server IP** setting, type the IP address for the remote storage server.
9. For the **Server Port** setting, type a port number or use the default value, **514**.
10. To ensure that the system logs requests for the web application, even when the logging utility is competing for system resources, check the **Guarantee Logging** box.

    ***Note**: Enabling this setting may slow access to the associated web application.*
11. Optionally, adjust the maximum request, header, and query string size and maximum entry length settings. (Refer to online help for details on the settings.)
12. If you want the system to log details (including the start and end time, number of dropped requests, attacking IP addresses, and so on) about brute force attacks, DoS attacks, IP enforcer attacks, or web scraping attacks, check the **Report Detected Anomalies** box.

13. In the Storage Filter area, make any changes as required. (See *Configuring the storage filter*, on page 14-10, for details.)
14. Click the **Create** button.
    The screen refreshes, and displays the new logging profile on the Logging Profiles screen.

Configuring a logging profile if using ArcSight logs

If your network uses ArcSight™ logs, you can configure a logging profile that formats the log information for that system. Application Security Manager stores all logs on a remote logging server using the predefined ArcSight settings for the logs.

The log messages are in Common Event Format (CEF). The basic format is:

```
CEF:Version|Device Vendor|Device Product|Device Version
|Device Event Class ID|Name|Severity|Extension
```

Note

This logging profile relies on external systems to perform the actual logging. The configuration and maintenance of the external logging servers is not the responsibility of F5 Networks.

To create a logging profile for ArcSight logs

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Logging Profiles**.
   The Logging Profiles screen opens.
2. Above the Logging Profiles area, click the **Create** button.
   The Create New Logging Profile screen opens.
3. For the **Configuration** setting, select **Advanced**.
   The screen refreshes to display additional settings.
4. For the **Profile Name** setting, type a unique name for the logging profile.
5. Check the **Remote Storage** box, and for the **Type** setting, select **ArcSight**.
   The screen displays additional settings.
6. If you do not want data logged locally as well as remotely, click to clear the **Local Storage** check box.
7. For the **Protocol** setting, select the protocol that the reporting server uses: **TCP** (the default setting), **UDP**, or **TCP-RFC3195**.
8. For the **Server IP** setting, type the IP address of the remote storage server.
9. For the **Server Port** setting, type a port number or use the default value, **514**.

10. To ensure that the system logs requests for the web application, even when the logging utility is competing for system resources, check the **Guarantee Logging** box.

    ***Note**: Enabling this setting may slow access to the associated web application.*

11. Optionally, adjust the maximum request, header, and query string size and maximum entry length settings. (Refer to online help for details on the settings.)

12. If you want the system to log details (including the start and end time, number of dropped requests, attacking IP addresses, and so on) about brute force attacks, DoS attacks, IP enforcer attacks, or web scraping attacks, check the **Report Detected Anomalies** box.

13. In the Storage Filter area, make any changes as required. (See *Configuring the storage filter*, following, for details.)

14. Click the **Create** button.
    The screen refreshes, and displays the new logging profile.

Configuring the storage filter

The storage filter of a logging profile determines the type of requests the system or server logs.

◆ Note

The following procedure describes configuring the storage filter for an existing logging profile.

To configure the storage filter

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Logging Profiles**.
   The Logging Profiles screen opens.

2. In the Logging Profiles area, click the name of an existing logging profile.
   The Edit Logging Profile screen opens.

3. For the **Storage Filter** setting, select **Advanced**.
   The screen refreshes to display additional settings.

4. For the **Logic Operation** setting, select the manner in which the system associates the criteria you specify. The criteria are the remaining settings in the storage filter.

   - **OR**: Select this operator if you want the system to log the data that meets one or more of the criteria.
   - **AND**: Select this operator if you want the system to log the data that meets all of the criteria.

5. For the **Request Type** setting, select the kind of requests that you want the system to store in the log.

6. For the **Protocols** setting, select whether logging occurs for HTTP and HTTPS protocols or a specific protocol.
7. For the **Response Status Codes** setting, select whether logging occurs for all response status codes or specific ones.
8. For the **HTTP Methods** setting, select whether logging occurs for all methods or specific methods.
9. For the **Request Containing String** setting, select whether the request logging is dependent on a specific string.
10. Click the **Update** button.
    The screen refreshes, and displays the new logging profile on the Logging Profiles screen.

Setting event severity levels for security policy violations

You can customize the severity levels of security policy violations for application security events that are displayed on the Security Alerts screen, in the request details, and also in the messages logged by the **syslog** utility. The event severity levels are **Informational**, **Notice**, **Warning**, **Error**, **Critical**, **Alert**, and **Emergency**. They range from least severe (**Informational**) to most severe (**Emergency**).

For more information on how BIG-IP systems use the **syslog** utility, refer to the *Logging BIG-IP System Events* chapter in the ***TMOS® Management Guide for BIG-IP® Systems***.

◆ Note

*When you make changes to the event severity level for security policy violations, the changes apply globally to **all** web applications.*

To customize event severity level for security policy violations

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **Severities**.
   The Severities screen opens.
2. For each violation, change the severity level as required.
3. Click the **Save** button to retain any changes.

◆ Tip

*If you modify the event severity levels for any of the security policy violations, and later decide you want to use the system-supplied default values instead, click the **Restore Defaults** button.*

Viewing the application security logs

Locally stored system logs for the Application Security Manager are accessible from the Configuration utility for the BIG-IP system. Note that these are the logs for general system events and user activity. Security violation events are displayed in the Configuration utility for the Application Security Manager.

To view logs that show all changes to security policies, refer to *Reviewing a log of all security policy changes*, on page 8-9.

For more information on logging in general, refer to the ***TMOS®*** ***Management Guide for BIG-IP® Systems***, which is available in the Ask F5SM Knowledge Base, **https://support.f5.com**.

◆ Tip

*If you prefer to review the log data from the command line, you can find the application security log data in the **/var/log/asm** directory.*

To view the application security logs

1. In the navigation pane of the BIG-IP system, expand **System**, and then click **Logs**.
   The System Logs list screen opens.
2. On the menu bar, click **Application Security**.
   The Application Security log list screen opens, where you can review the logged entries.

Validating regular expressions

The RegExp Validator is a system tool designed to help you verify your regular expression syntax. You can type a regular expression in the RegExp Validator, provide a test string pattern, and let the tool analyze the data.

To validate regular expressions

1. In the navigation pane, expand **Application Security**, point to **Options, Tools**, and then click **RegExp Validator**.
   The RegExp Validator screen opens.
2. In the **RegExp** box, perform one of the following tasks to specify how you want the validator to work:
   - Type the regular expression you want to validate.
   - Type the regular expression to use to verify a test string, and then in the **Test String** box, type the string.
3. Click the **Validate** button.
   The screen refreshes and shows the results of the validation.

Configuring an SMTP mail server

If you want the system to send email to users, such as when configuring the system to send reports using email (refer to *Scheduling and sending graphical charts using email*, on page 15-11), you must enable the SMTP mailer and configure an SMTP server.

◆ Note

*For the SMTP mailer to work, you must make sure the SMTP server is on the DNS lookup server list, and configure the DNS server on the BIG-IP system (**System>>Configuration>>Device>>DNS**).*

To configure SMTP

1. In the navigation pane, expand **Application Security**, point to **Options**, and then click **SMTP Configuration**.
   The SMTP Configuration screen opens.
2. Check the **Enable SMTP mailer** box.
3. For **SMTP Server Host Name**, type the fully qualified host name of an SMTP server (for example, **smtp.example.com**).
4. For **SMTP Server Port Number**, type the SMTP port number (**25** is the default for no encryption; **465** is the default if SSL or TLS encryption is the encryption setting).
5. For **Local Host Name**, type the fully qualified host name of the BIG-IP system.
6. For **From Address**, type the mail address to use as the reply-to address of the email.
7. For **Encrypted Connection**, select whether the SMTP server requires an encrypted connection to send mail. Select **No encryption**, **SSL** (Secure Sockets Layer), or **TLS** (Transport Layer Security).
8. If you want the SMTP server to validate users before sending email, check the **Use Authentication** box, then type the **Username** and **Password** that the SMTP server requires for validation.
9. Click **Save** to save the configuration.

15

Displaying Reports

- Overview of the reporting tools
- Displaying an application security overview
- Reviewing details about requests
- Viewing charts
- Scheduling and sending graphical charts using email
- Viewing anomaly statistics
- Viewing PCI Compliance reports
- Filtering reports
- Monitoring CPU usage

Overview of the reporting tools

You can use several reporting tools in Application Security Manager™ to analyze incoming requests, track trends in violations, generate security reports, and evaluate possible attacks. The statistics and monitoring reporting tools are:

- **Application security overview**
  Displays a summary of all configured web applications showing the active security policies, attack types that have occurred, anomaly statistics, and networking and traffic statistics. See *Displaying an application security overview*, on page 15-2, for details.
- **Requests summary**
  Summarizes the requested URLs for web applications. See *Reviewing details about requests*, on page 15-4, for more information.
- **Charts**
  Displays graphical reports about security policy violations and provides tools that let you view the data by different criteria, drill down for more data, create customized reports, and export reports. See *Viewing charts*, on page 15-8, for more information.
- **Charts Scheduler**
  Allows you to periodically generate specific reports and distribute them using email.
- **DoS Attacks report**
  Displays DoS attack events, listed by the web application targeted, and the attack start and end times. See *Viewing DoS Attacks reports*, on page 15-12, more information.
- **Brute Force Attacks report**
  Displays brute-force attack events, including the web application attacked, login URL, and attack start and end times. See *Viewing Brute Force Attack reports*, on page 15-13, for more information.
- **IP Enforcer Statistics**
  Lists the IP addresses containing requests that exceeded the maximum number of blocked violations, and you can see additional details about the request and associated violations.
- **Web Scraping Statistics**
  Displays details about web scraping attacks that the system detected and logged.
- **PCI Compliance report**
  Displays a printable Payment Card Industry (PCI) compliance report for each web application showing each security measure required for PCI-DSS 1.2, and compliance details.

Displaying an application security overview

You can display an overview where you can quickly see what is happening on the Application Security Manager including:

- Web applications, state, active security policies, and policy building status
- Attack type statistics
- Anomaly detection statistics
- Traffic summary showing blocked or dropped requests
- Networking statistics showing transactions per second or throughput
- Top requested URLs
- Top requesting IP addresses

To display overview statistics

1. In the navigation pane, expand **Application Security**, and click **Overview**.
   The Overview screen opens and summarizes system activity at a glance.
2. In the Statistics area, from the **Web Application** list, select an application to narrow down the statistics. By default, statistics for all web applications are displayed.
3. To specify how far back you want to view the statistics, after **Time Period**, click **Last Hour**, **Last Day**, or **Last Week**.
4. Review the summary statistics to determine what is happening on the system.

   ***Tip**: See the online help for details about the tables and graphs.*

Figure 15.1 shows what the Application Security Overview screen (top part) looks if attacks have occurred, with a pie chart showing the types of attacks. The bottom of the screen includes additional traffic and networking statistics that you can scroll down to see.

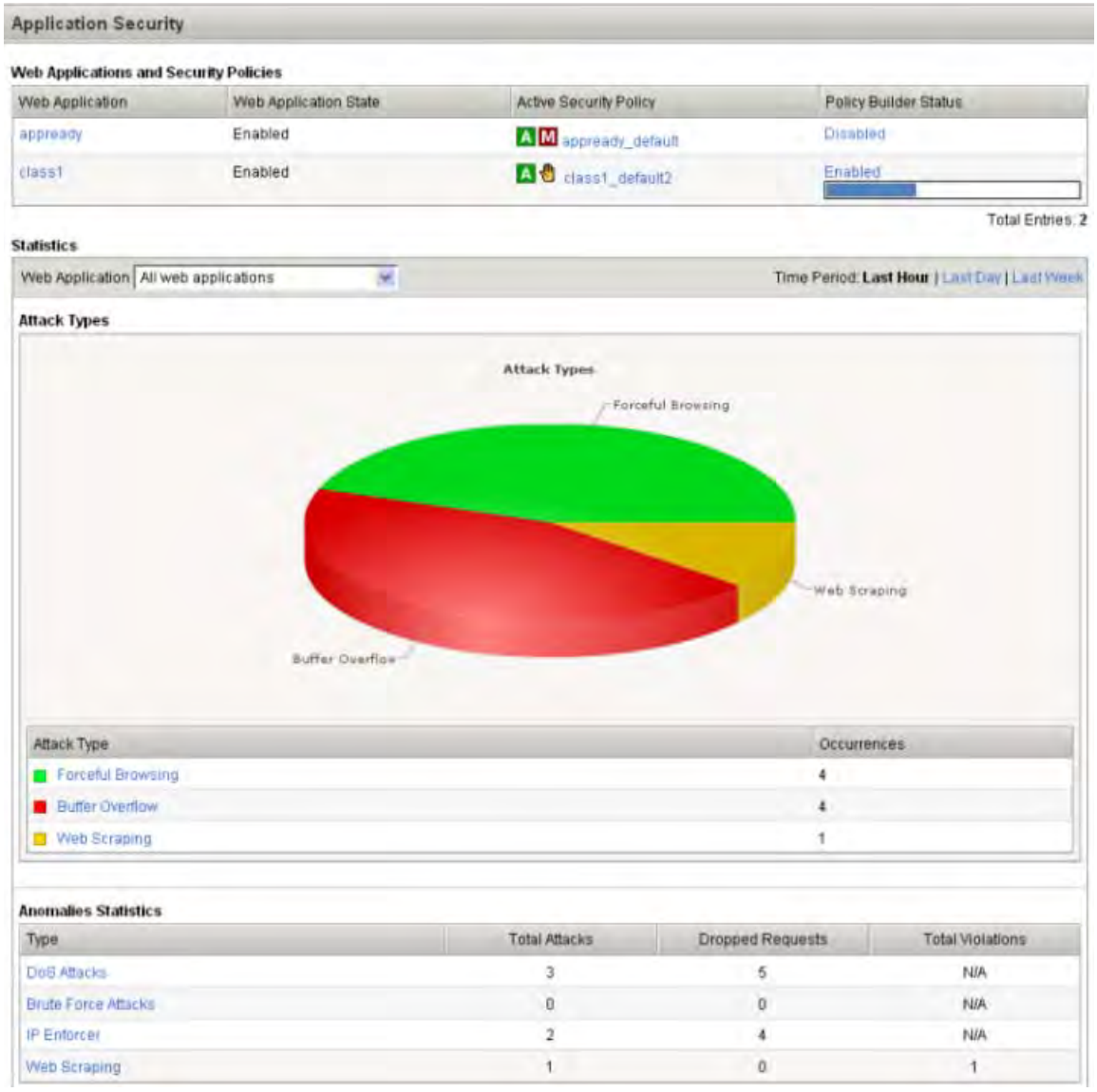


Figure 15.1 *Application Security overview statistics*

Reviewing details about requests

For each web application, the Application Security Manager logs requests according to the logging profile (**Options** >> **Logging Profiles**). If you use local logging, you can review those requests in the Requests List on the Requests screen. For more information on configuring logging profiles, refer to *Configuring logging profiles for web application data*, on page 14-5.

The Requests List provides information about a request such as: the request category, the time of the request, its severity, the source IP address of the request, the server response code, and the requested URL itself, as shown in Figure 15.2. Icons on each request line provide additional status information such as whether the request is legal or illegal, blocked, truncated, or has a response. The request legend describes these icons.

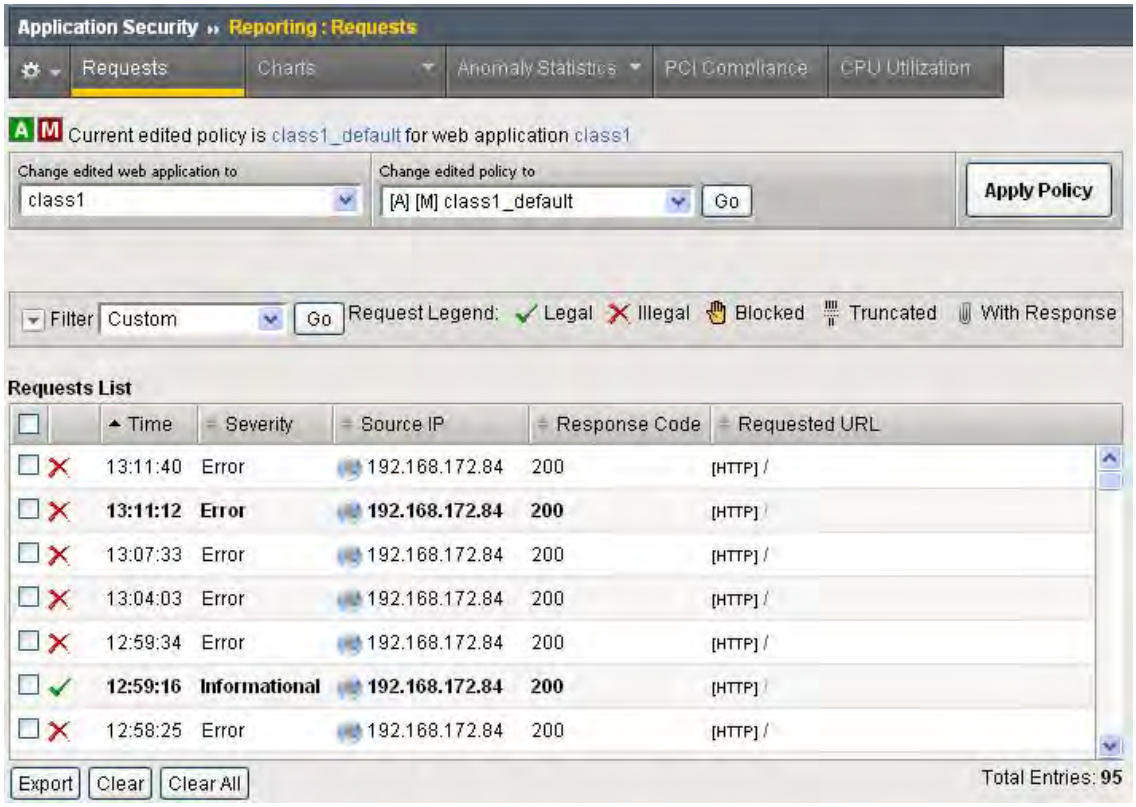


Figure 15.2 *Request list and legend*

You can view additional details about a request, including viewing the full request itself, and any violations associated with it. You can also drill down to view detailed descriptions of the violations and potential attacks.

When viewing details about an illegal request, if you decide that the request is trusted and you want to allow it, you can accept the violations shown for this specific request.

You can use a filter to view only those requests and events that are of interest to you, as described in *Filtering reports*, on page 15-17. The filter list has several built-in options that you can use to display all requests, legal requests, illegal requests, or requests that occurred within a certain time range. You can also create a custom filter and view requests by attack type, source IP address, HTTP method used, and many other options.

To view details about requests and violations

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens, where you can review a list of requests for all web applications.

   Note: *If* ***Filter*** *details are displayed, the Requests List appears below them.*

2. In the Requests List, click anywhere on a request if you want to view information about the request and any violations associated with it.
   - Click the link in the Requested URL column to display details in a separate popup screen.
   - Click elsewhere on the line to display details on the same screen, below the Requests List. If later you want to hide the details, click the heading line.

   Either place, you see any violations associated with the request and other details, such as the web application it relates to, the support ID, severity, and potential attacks that it could cause. As an example, Figure 15.3 shows information about a request that caused two potential violations.

3. To view more details about a violation:
   - Click the icon to the left of the violation to display a general description of that type of violation. As an example, Figure 15.4 shows the description of the **Illegal header length** violation.
   - If the violation is set to **Alarm** or **Block**, click the violation name to view details about this specific violation such as the file type, the expected and actual length of the query, or similar relevant information. In the popup screen that appears, you see additional details, and, for attack signatures, you can click **View details** to get context details.

4. For violations that you want to allow (false positives), click the **Learn** button.
   If there are learning suggestions, the violation's learning screen opens where you can accept the suggestions one at a time.

5. To view the actual content of the request, click **Full Request**.
   The content of the full request replaces the list of violations.

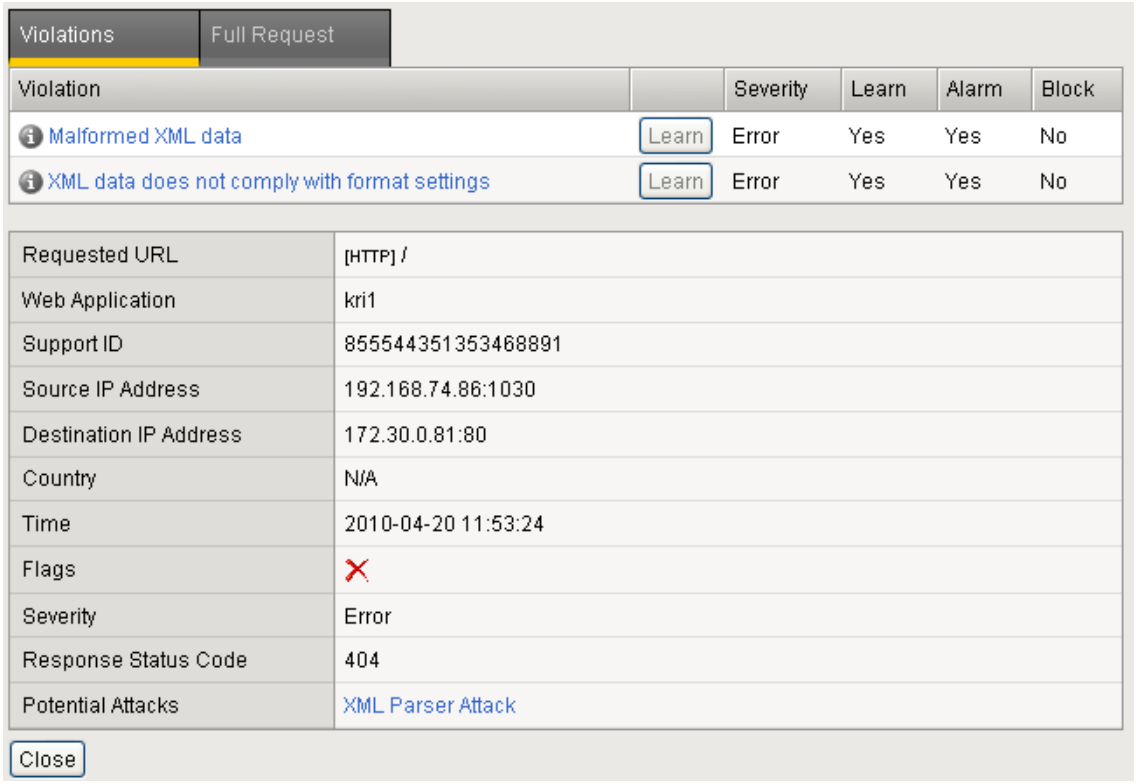


Figure 15.3 *Request details*

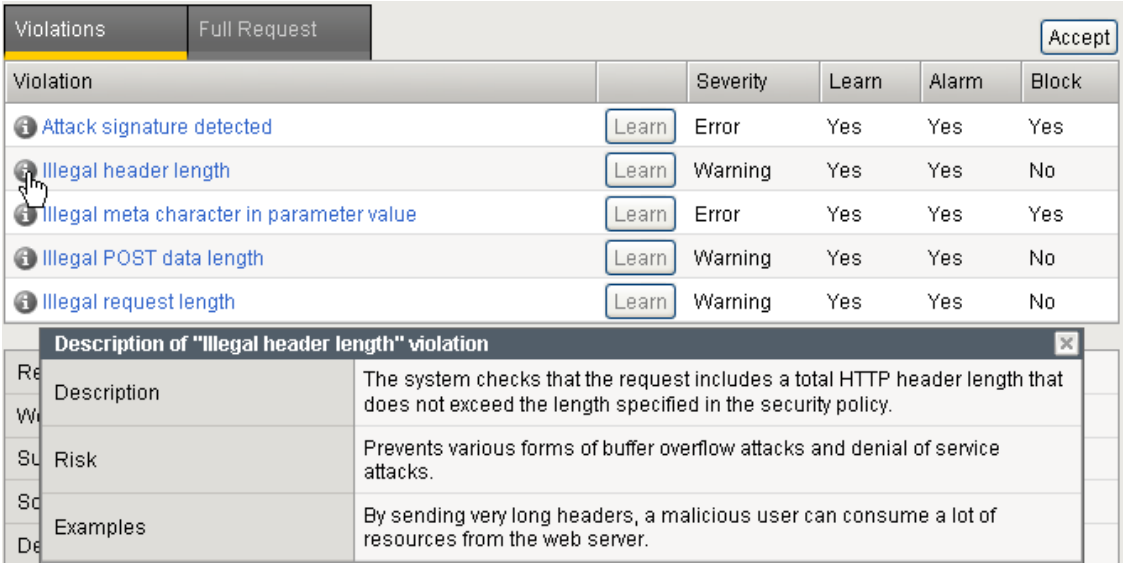


Figure 15.4 *Details about Illegal header length violation*

Exporting requests

You can export selected requests in PDF or binary format for troubleshooting purposes.

To export requests

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens.
2. If you want to export specific requests, select those requests from the list. You can export up to 100 entries in PDF format.
3. At the bottom of the Requests List, click **Export**.
   The Select Export Method popup screen provides options.
4. Select the export method to use, then click **Export**:
   - To export selected requests into a document, click **Export selected requests in PDF format**.
     You can choose to open or save the file created.
   - To export requests into a document and send it by e-mail, click **Send selected requests in PDF format to your E-mail address**, and type your e-mail address.

     Note: *To use this option, you must first enable the SMTP mailer as described in* ***Configuring an SMTP mail server****,* on page 14-14.
   - To export all requests to a tar file, click **Binary export of all requests defined by filter**.
     The system creates a ***.tar.gz** file of the requests, and saves it where you specify.

Clearing requests

If you have reviewed and dealt with requests, you may want to clear them from the Requests List. This is an optional task.

To clear requests from the Requests List

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens.
2. Select which requests to clear:
   - To clear selected requests, select the requests to delete and click **Clear**.
   - To clear all requests, click **Clear All**.

   The systems prompts you to confirm the deletion, then removes the requests from the Requests List without changing the security policy.

Viewing charts

You can display numerous graphical charts that illustrate the distribution of security alerts. You can filter the data by web application and time period, and you can view illegal requests based on different criteria such as web applications, violations, attack types, URLs, IP addresses, severity, response codes, request types, or protocols.

The system provides several predefined filters that produce charts focused on areas of interest including the top alerted applications, top violations, top attacks, and top attackers. You can use these charts as executive reports that summarize your overall system security.

You can also send charts to people periodically using email; for details, see *Scheduling and sending graphical charts using email*, on page 15-11.

Figure 15.5 is an example of a chart that shows the violations that have occurred on the system. Details below the chart include the number of occurrences for each type of violation.

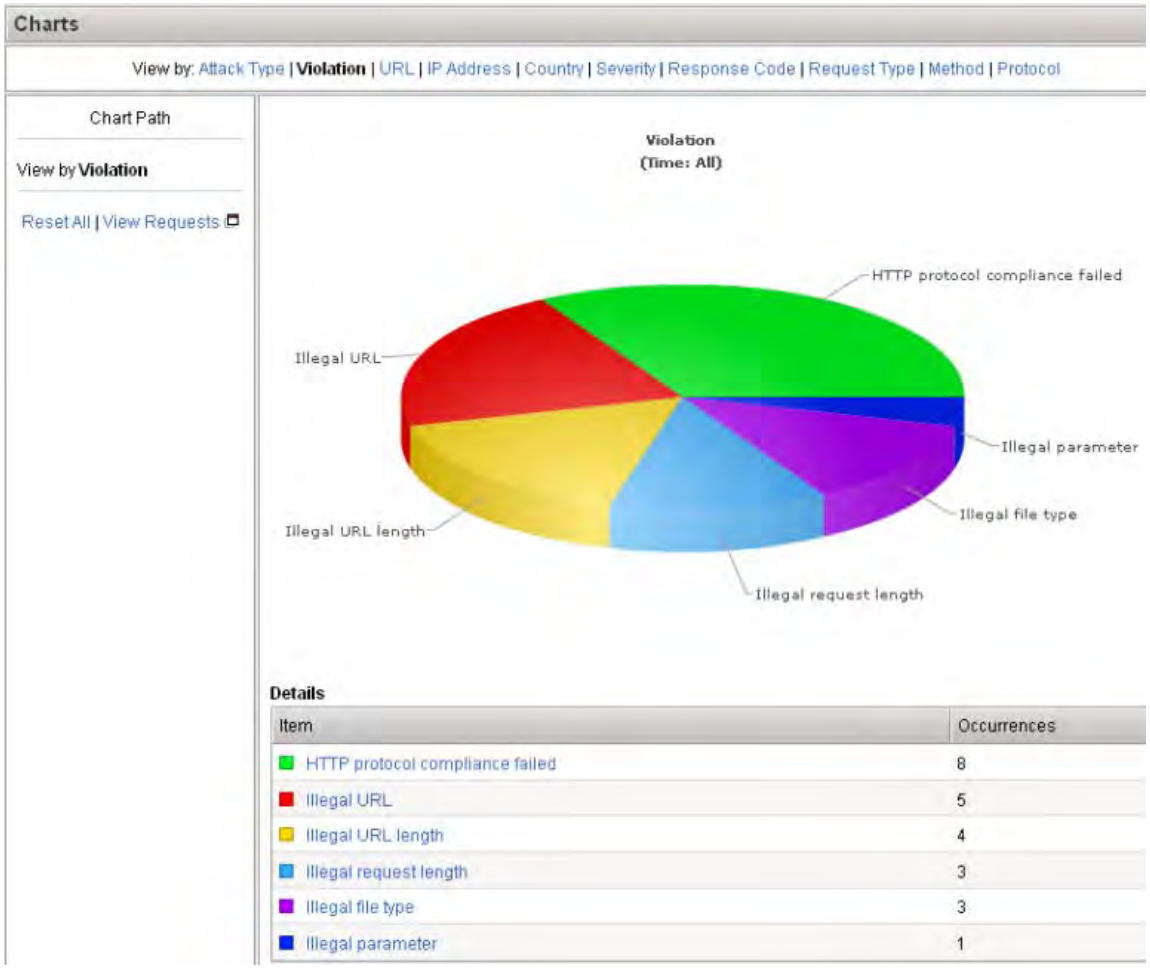


Figure 15.5 *Chart showing attacks by violation*

You can use a filter to view the security incidents which are of interest to you. The filter list has several predefined options. In addition, you can create a custom filter. See *Filtering reports*, on page 15-17.

The easiest way to learn about the graphical reports is to display a report, then change the view by criteria, and drill down into the report to display details about particular aspects you are interested in. The different steps you take are shown in the Chart Path on the left of the screen.

To view graphical charts

1. In the navigation pane, expand **Application Security**, click **Reporting**.
   The Requests screen opens.
2. From the Charts menu, choose Charts.
   The Charts screen opens, where you can view graphical reports.
3. From the **Filter** list, select the predefined or custom filter you want to use and click **Go**. For details, see *Filtering reports*, on page 15-17.
4. In the Charts section, next to **View by**, click the viewing criteria for the report you want to see.
   The Reports screen displays a graphical report of illegal requests by the selected criteria. For example, if you selected view by **Violation**, the report shows each type of violation against the security policy in a pie chart (shown previously in Figure 15.5), followed by a details table, and a bar chart, which displays the violations that occurred over time.
5. Click any slice in the pie chart or detail in the details table to display more information about that specific item.
   The graphical report shows more details, and the view by choices are relevant only to the selection you made. For example, if viewing by **Attack Type**, you can click any attack type to view how many attacks of this type occurred for each application.
6. Change the drilldown settings in the Chart Path as needed:
   - Click the close button (⊠) to close an item in the drilldown path and change the report view.
   - Click **Reset All** to remove all drilldown settings for the report but keep the view by criteria.
   - Click **View Requests** to view the requests that relate to the current report.
7. To create a PDF version of the report that you can save or print (including charts based on your drill downs), at the bottom of the screen, click **Export**.
   The system asks if you want to open or save the PDF file.

Interpreting graphical charts

You can monitor graphical charts to determine how well your security policies are protecting your web applications. By viewing specific charts, you can check for false positives and adjust security policies accordingly. The contents of the charts can help you to determine why the system flagged certain requests as illegal.

For example, if you notice that many attacks are emanating from one IP address, you have identified a possible attacker. You can check the validity of that IP address. You may want to enable session-based enforcement to block those requests producing too many violations and coming from a single IP address. See *Configuring IP address enforcement*, on page 7-12, for more information.

If you see that the same type of attack is coming from many different IP addresses, this may indicate a false positive, and you may need to adjust your security policy. As an example, if you see many illegal URL violations and find that they are coming from many different IP addresses, you should consider adding this URL to the security policy.

By viewing graphical reports periodically and investigating the illegal requests using different criteria, you can evaluate system vulnerabilities. As you get more familiar with the report details, you can use the information that you get to further secure your application traffic.

Scheduling and sending graphical charts using email

You can configure the Charts Scheduler to send predefined and customized charts to specific email addresses periodically. Create a schedule for each chart that you want to send. Figure 15.6 shows the an example of the chart scheduler.

◆ Note

*You must configure SMTP before you can send email notifications. If SMTP is not configured, an alert appears on the screen that links to SMTP configuration (**Options>>SMTP Configuration**). Also, make sure the SMTP server is on the DNS lookup server list, and configure the DNS server that you want the system to use (**System>>Configuration>> Device>>DNS**).*

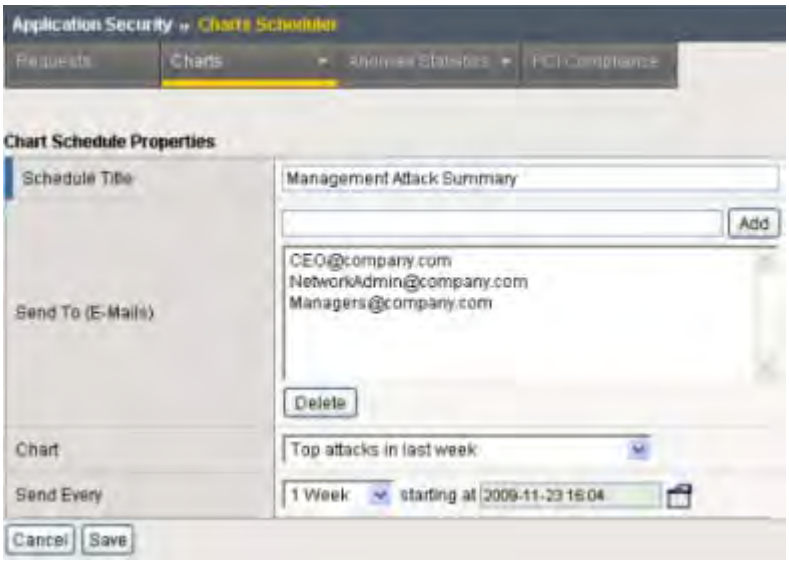


***Figure 15.6** Sending charts by email using the chart scheduler*

To schedule sending a chart by email

1. In the navigation pane, expand **Application Security**, point to **Reporting,** then **Charts**, and click **Charts Scheduler**.
   The Charts Scheduler screen opens.
2. If you see an SMTP alert, click the setup link and configure SMTP (as described in *Configuring an SMTP mail server*, on page 14-14). Otherwise, skip this step.
3. Click the **Create** button.
   The Chart Schedule Properties screen opens.
4. For **Schedule Title**, type a name for this schedule.

5. In the **Send To (E-Mails)** box, type each email address where you want the system to send a copy of the chart, then click **Add**.
6. From the **Chart** list, select the predefined chart to send.
7. For **Send Every**, select how often to send the charts, and after **starting at**, set the time and date to begin sending the charts.
8. Click **Create** to save the schedule.
   The Chart Scheduler screen shows the schedule you added.

Viewing anomaly statistics

You can view reports showing DoS attacks, brute force attacks, IP Enforcer statistics, and web scraping statistics.

Viewing DoS Attacks reports

The DoS Attacks report displays information about denial of service (DoS) attacks, including the associated web application and the start and end times of an attack. For details on configuring DoS attack detection, see *Preventing DoS attacks for Layer 7 traffic*, on page 7-2.

To view all DoS attacks

1. In the navigation pane, expand **Application Security**, point to **Reporting**, **Anomaly Statistics**, then click **DoS Attacks**.
   The DoS Attacks screen opens.
2. From the **Filter** list, select **Show All.**
3. Click **Go**.
   The screen refreshes, and the DoS Attacks report displays all DoS attack events.
4. To view statistical details about a DoS attack, click the **View** button or **View** link in the Details column.
   The system displays details it has collected about the attack, such as latency history and end time, dropped connections per IP address and URL, mitigation, IP addresses of the attackers, and attacked URLs.
5. You can filter the report to display only those events you are interested in. For details, see *Filtering reports*, on page 15-17.

Viewing Brute Force Attack reports

The Brute Force Attack report displays information about brute force attacks, including the web application, login URL, and start and end times of an attack. For details on configuring brute force attack detection, see *Mitigating brute force attacks*, on page 7-6.

To view all brute force attacks

1. In the navigation pane, expand **Application Security**, point to **Reporting**, **Anomaly Statistics**, then click **Brute Force Attacks**. The Brute Force Attacks screen opens.
2. From the **Filter** list, select **Show All.**
3. Click **Go**.
   The screen displays a report to show all brute force attack events.
4. You can filter the report to display only those events you are interested in. For details, see *Filtering reports*, on page 15-17.

Viewing IP Enforcer statistics

The IP Enforcer statistics are available in the **Reporting** section of the Application Security Manager. The IP Enforcer Statistics report shows the IP addresses of the clients that were attacking a web application, and which requests were blocked based on a security policy and IP Enforcer configuration. For details about the IP Enforcer, see *Configuring IP address enforcement*, on page 7-12.

◆ Note

*To gather IP Enforcer statistics, you must have configured the IP Enforcer in the **Blocking** or **Transparent** operation mode, and the security policy must be in **Blocking** enforcement mode and must block one or more violations.*

To view all IP Enforcer statistics

1. In the navigation pane, expand **Application Security**, point to **Reporting**, **Anomaly Statistics**, then click **IP Enforcer Statistics**. The IP Enforcer Statistics screen opens.
2. From the **Filter** list, select **Show All.**
3. Click **Go**.
   The IP Enforcer Statistics screen displays all IP Enforcer statistics.
4. You can filter the report to display only those statistics you are interested in. For details, see *Filtering reports*, on page 15-17.

To release IP addresses blocked by the IP Enforcer

1. In the navigation pane, expand **Application Security**, point to **Reporting**, **Anomaly Statistics**, then click **IP Enforcer Statistics**.
   The IP Enforcer Statistics screen opens.
2. Select the client IP addresses that you want to unblock, and click **Release**.
   The system considers the attack from this IP address to be over, and puts the time you released the IP address in the End Time column. The IP address remains in the list of IP Enforcer statistics.

Viewing web scraping statistics

The Web Scraping Statistics report displays information about web scraping attacks that the system detected and logged. The statistics include the client IP address, web application, start and end time, and the number of dropped and violating requests. For details on configuration web scraping detection, see *Detecting and preventing web scraping*, on page 7-13.

Figure 15.7 shows an example of web scraping statistics that all originate from the IP address **192.168.172.60** for the web application called **asas**.

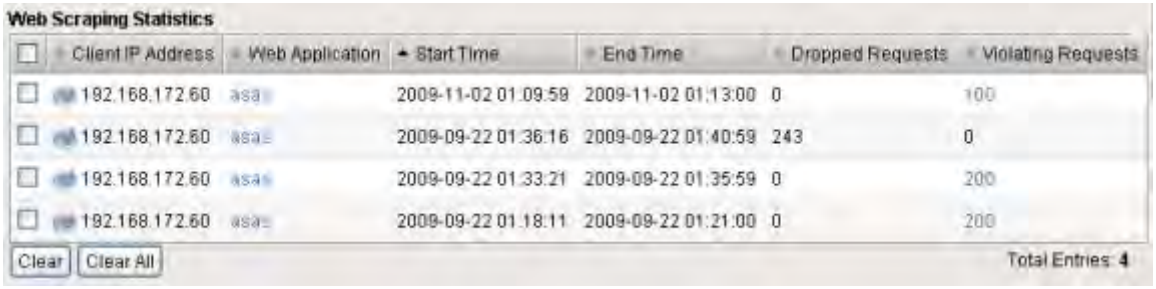


Figure 15.7 *Example of web scraping statistics*

To view all web scraping statistics

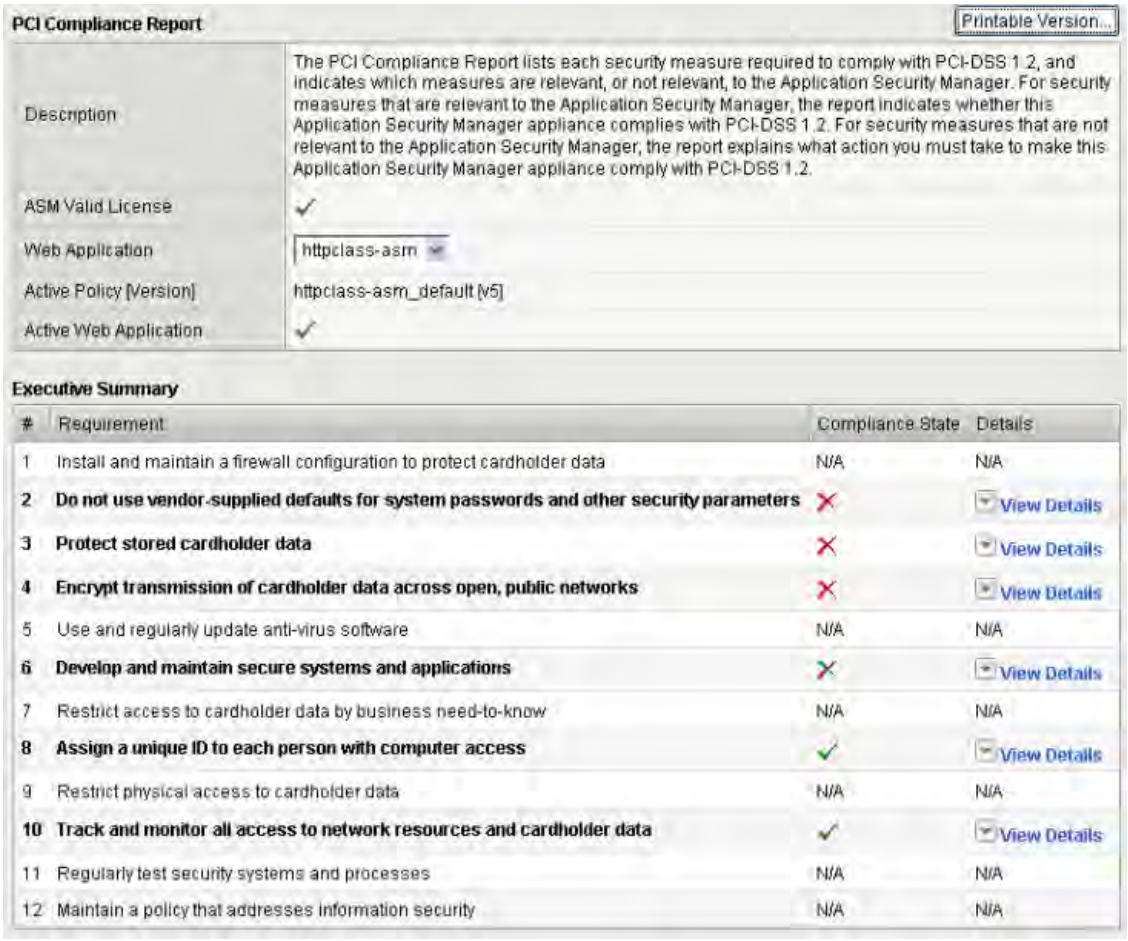
1. In the navigation pane, expand **Application Security**, point to **Reporting**, **Anomaly Statistics**, then click **Web Scraping Statistics**.
   The Web Scraping Statistics screen opens.
2. From the **Filter** list, select **Show All.**
3. Click **Go**.
   The screen refreshes, and the Web Scraping Statistics displays all incidents of web scraping that were detected.
4. You can filter the report to display only those events you are interested in. For details, see *Filtering reports*, on page 15-17.

Viewing PCI Compliance reports

The PCI Compliance report displays details on how closely the security policy of a web application meets Payment Card Industry (PCI) security standards, PCI-DSS 1.2. The report indicates which requirements Application Security Manager can help enforce, and allows you to view details about what to configure differently to meet compliance standards. The PCI Compliance report shows the configuration of the active security policy, if the web application has two or more security policies associated with it.

You can create printable versions of PCI compliance reports for each web application to assure auditors that the BIG-IP system and your web applications are secure.

Figure 15.8 shows a sample PCI Compliance report with two requirements in compliance, three not in compliance, one partially compliant, and several items that you must make compliant outside of Application Security Manager. Note that fixing items outside of Application Security Manager does not change the compliance state in the report.



PCI Compliance Report — Printable Version...

Description	The PCI Compliance Report lists each security measure required to comply with PCI-DSS 1.2, and indicates which measures are relevant, or not relevant, to the Application Security Manager. For security measures that are relevant to the Application Security Manager, the report indicates whether this Application Security Manager appliance complies with PCI-DSS 1.2. For security measures that are not relevant to the Application Security Manager, the report explains what action you must take to make this Application Security Manager appliance comply with PCI-DSS 1.2.
ASM Valid License	✓
Web Application	httpclass-asm
Active Policy [Version]	httpclass-asm_default [v5]
Active Web Application	✓

Executive Summary

#	Requirement	Compliance State	Details
1	Install and maintain a firewall configuration to protect cardholder data	N/A	N/A
2	**Do not use vendor-supplied defaults for system passwords and other security parameters**	✗	View Details
3	**Protect stored cardholder data**	✗	View Details
4	**Encrypt transmission of cardholder data across open, public networks**	✗	View Details
5	Use and regularly update anti-virus software	N/A	N/A
6	**Develop and maintain secure systems and applications**	✗	View Details
7	Restrict access to cardholder data by business need-to-know	N/A	N/A
8	**Assign a unique ID to each person with computer access**	✓	View Details
9	Restrict physical access to cardholder data	N/A	N/A
10	**Track and monitor all access to network resources and cardholder data**	✓	View Details
11	Regularly test security systems and processes	N/A	N/A
12	Maintain a policy that addresses information security	N/A	N/A

Figure 15.8 *Example PCI Compliance report*

To view a PCI Compliance report

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens.
2. On the menu bar, click **PCI Compliance**.
   The PCI Compliance Report screen opens showing a compliance report for the current web application.
3. To learn more about items that are PCI compliant (items with a green ✔) or those which are not PCI compliant (items with a red **X**), in the Details column, click the **View Details** link.
   The screen shows information about how to make an item compliant. For example, Figure 15.9 shows that vendor-supplied default passwords are used for the **root** and **admin** users.
4. To create a PDF version of the report that you can save, or open and print, click **Printable Version**.
5. To display a PCI compliance report for a different web application, from the **Web Application** list, select the web application name.
   A PCI compliance report for the new web application opens.

Executive Summary
Requirement Compliance State Details
1 Install and maintain a firewall configuration to protect cardholder data N/A N/A
2 Do not use vendor-supplied defaults for system passwords and other security parameters
Hide details
Default Users
Username Default password is used
root Yes
admin Yes

Figure 15.9 *PCI Compliance details*

Filtering reports

You can use a filter to view the information of interest to you in several of the reports. You can filter reports that show requests, charts, and anomaly statistics.

You can use the predefined filter options that are applicable to each type of information. Alternately, you can create a custom filter that refines the report by criteria such as web application and time period.

To use a predefined filter

1. In the navigation pane, expand **Application Security**, click **Reporting** and then display the report you are interested in.
2. From the **Filter** list, select the filter you want to use.
3. Click **Go**.
   The screen displays the filtered report.

To create a custom filter

1. In the navigation pane, expand **Application Security**, click **Reporting** and then display the report you are interested in.
2. If the filter options are not displayed, to the left of the **Filter** setting, click the Show/Hide Filter button ().
   The screen displays the custom filter options.
3. Specify the criteria by which you want the **Filter** option to filter the report. The filter options vary for different reports. Refer to the online help for descriptions of the options.
4. Click the **Save Filter** button.
   A popup screen opens.
5. Type a name for the custom filter, and click **OK**.
   The screen refreshes, and you see the custom filter in the **Filter** list.
6. From the **Filter** list, select the custom filter that you just created, and then click **Go**.
   The report displays the filtered information.

Monitoring CPU usage

You can examine the amount of CPU resources that the Application Security Manager is using, and also check overall system CPU usage.

To check CPU usage for Application Security Manager

- In the navigation pane, expand **Application Security**, point to **Reportin**g and click **CPU Utilization**.
  The CPU Utilization screen opens.

To check overall system CPU usage

- In the navigation pane, expand **Overview** and click **Performance**.
  On the Performance screen that opens, you can view system CPU usage.

A

Security Policy Violations

- Introducing security policy violations
- Viewing descriptions of violations
- RFC violations
- Access violations
- Length violations
- Input violations
- Cookie violations
- Negative security violations
- Filtering requests by attack type

Introducing security policy violations

Security policy violations (or just violations) occur when some aspect of a request or response does not comply with the security policy for a web application. This appendix describes each of the violations.

Viewing descriptions of violations

You can view detailed descriptions of each violation to learn what causes that type of violation, and the type of security risk it could be related to.

To view descriptions of violations

1. In the navigation pane, expand **Application Security**, point to **Policy**, then **Blocking**, and then click **Settings**.
   The Blocking Policy screen opens listing all the violations and blocking settings for the current policy.

2. Click the icon preceding the violation you are interested in.
   A popup screen shows the violation description, risks, and examples, if available.

Figure A.1 shows a portion of the blocking policy screen, and Figure A.2 shows the description that you see when you click the icon for the **Illegal file type** violation.

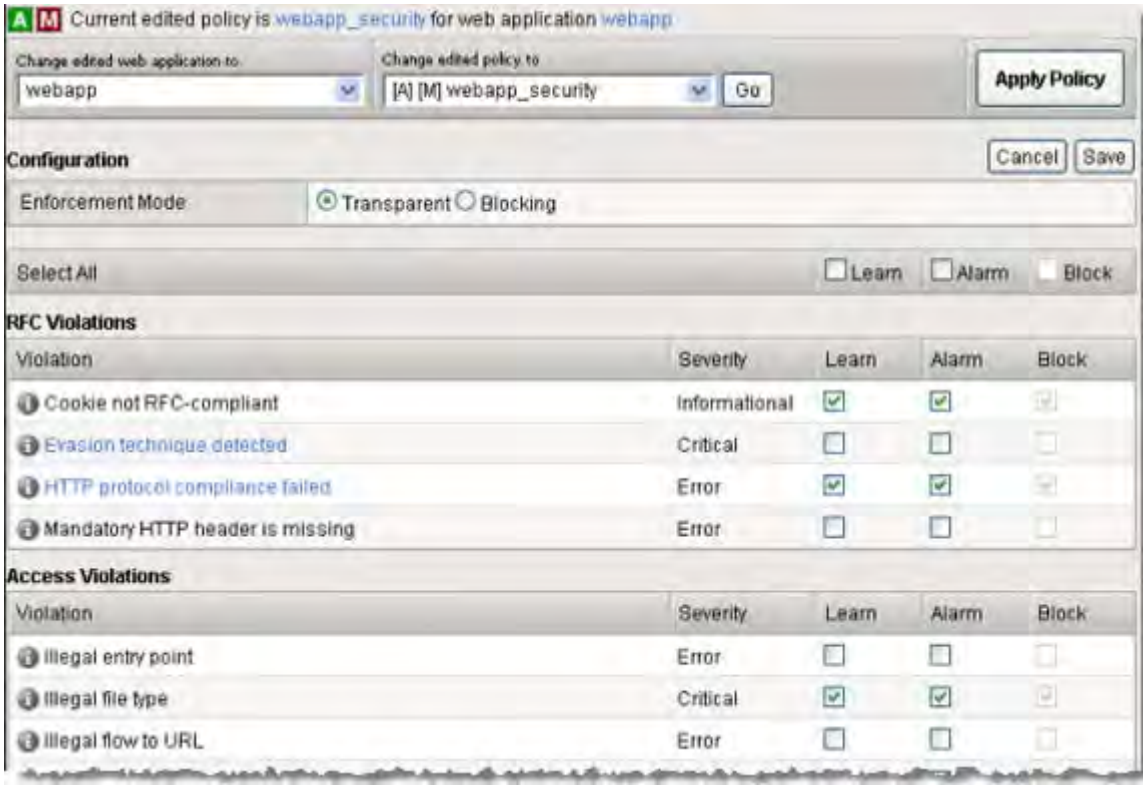


Figure A.1 *Violations on Blocking Policy screen*

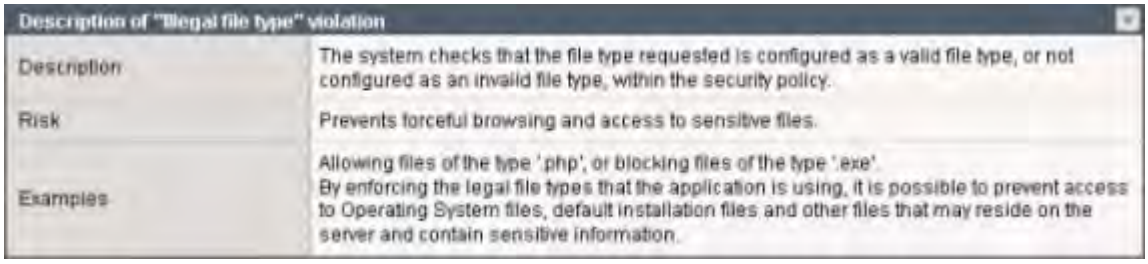

Description of "Illegal file type" violation
Description
The system checks that the file type requested is configured as a valid file type, or not configured as an invalid file type, within the security policy.
Risk
Prevents forceful browsing and access to sensitive files.
Examples
Allowing files of the type '.php', or blocking files of the type '.exe'.
By enforcing the legal file types that the application is using, it is possible to prevent access to Operating System files, default installation files and other files that may reside on the server and contain sensitive information.

Figure A.2 *Example violation description*

Many violations are associated with an attack type, and you can filter attack signatures or illegal requests by attack type (for more information, see *Creating a custom filter for attack signatures*, on page 11-7 and *Filtering requests by attack type*, on page A-13). Some violations are caused by multiple types of attack and do not have one attack type associated with them.

RFC violations

The Application Security Manager™ reports RFC violations when the format of an HTTP request violates the HTTP RFCs. ***RFC documents*** are the general specifications that summarize the standards used across the Internet and networking engineering community. RFCs, as they are commonly known, are published by the International Engineering Task Force (IETF). For more information on RFCs, see **http://www.ietf.org/rfc**.

Table A.1 lists the RFC violations, describes the event that triggers the violation, and specifies the attack type (if one is associated with the violation).

RFC violation	Violation trigger event	Attack type
Cookie not RFC-compliant	The cookie header in the request does not comply with the formatting standards as specified in the RFC for HTTP state management.	HTTP parser attack
Evasion technique detected	The content of the request contains encoding or formatting that represents an attempt to bypass attack signature detection. The following subviolation checks can occur:	Depends on subviolation
	Directory traversals The request includes directory traversal commands such as ../	Path traversal
	Multiple decoding **n** decoding passes The URI or parameter values are encoded multiple times and may indicate an attack. You can set the number of decoding passes (**2**, **3**, **4**, or **5**) at which to issue the violation.	Detection evasion
	%u decoding The system performs Microsoft %u unicode decoding to check for various attacks.	Detection evasion
	IIS backslashes The system normalizes backslashes to slashes to prevent attackers from requesting files.	Detection evasion
	IIS Unicode codepoints The system handles the mapping of IIS-specific non-ASCII codepoints.	Detection evasion
	Bare byte decoding The system detects ASCII bytes higher than 127.	Detection evasion
	Apache whitespace The system detects the following characters in the URI: 0x09, 0x11, 0x12, and 0x13.	Detection evasion

Table A.1 *RFC violations*

RFC violation	Violation trigger event	Attack type
	Bad unescape The system detects illegal HEX encoding and reports unescaping errors (such as %RR).	Detection evasion
HTTP protocol compliance failed	The request does not comply with one of the following HTTP protocol compliance checks:	Depends on subviolation
	POST request with Content-Length: 0	HTTP Request Smuggling Attack
	Header name with no header value	None
	Several Content-Length headers	HTTP Request Smuggling Attack
	Chunked request with Content-Length header	None
	Body in GET or HEAD requests	None
	Bad multipart/form-data request parsing	HTTP Parser Attack
	Bad multipart parameters parsing	HTTP Parser Attack
	No Host header in HTTP/1.1 request	Non-browser client
	CRLF characters before request start	None
	Host header contains IP address	None
	Content length should be a positive number	HTTP Parser Attack
	Bad HTTP version	Non-browser client
	Null in request	Injection Attempt
	High ASCII characters in headers	None
	Unparsable request content	HTTP Parser Attack
	Check maximum number of headers: n maximum headers	HTTP Parser Attack
Mandatory HTTP header is missing	The request does not contain an HTTP header specified as mandatory by the security policy.	None

Table A.1 *RFC violations (Continued)*

Access violations

Access violations occur when an HTTP request tries to gain access to an area of a web application, and the system detects a reference to one or more entities that are not allowed (or are specifically disallowed) in the security policy. Table A.2 lists the access violations, describes the event that triggers the violation, and specifies the attack type (if one is associated with the violation).

Access violation	Violation trigger event	Attack type
CSRF attack detected	The request is not legitimate and comes from a clicked link, embedded malicious HTML, or JavaScript in another application, and may involve transmission of unauthorized commands through an authenticated user. Cross-Site Request Forgery (CSRF) is suspected.	None
CSRF authentication expired	The system injects a CSRF session cookie into responses. If you configured an expiration time for CSRF protection, and the request was sent after the CSRF session cookie expired, the system issues this violation.	None
Illegal entry point	The incoming request references a URL that is not defined as an entry point.	Forceful browsing
Illegal file type	The incoming request references a file type that is not specified on the allowed file types list or is specified on the disallowed file types list in the security policy.	Forceful browsing
Illegal flow to URL	The incoming request references a flow that is not found in the security policy.	Forceful browsing
Illegal HTTP status in response	The server response contains an HTTP status code that is not defined in the security policy.	None
Illegal meta character in parameter name	The incoming request includes a parameter that contains a meta character that is not allowed in the security policy.	None
Illegal meta character in URL	The incoming request includes a URL that contains a meta character that is not allowed in the security policy.	None
Illegal method	The incoming request references a HTTP method that is not defined in the security policy.	Information leakage
Illegal session ID in URL	The system checks that the request contains a session ID value that matches the session ID value that the server set for this session.	Session hijacking

Table A.2 *Access violations*

Access violation	Violation trigger event	Attack type
Illegal URL (also called Non-existent URL)	The incoming request references a URL that is not specified on the allowed URLs list or is specified on the disallowed URLs list in the security policy.	Forceful browsing
Login URL bypassed	The incoming request tried to access the web application without going through the login URL.	Forceful browsing
Login URL expired	The incoming request is for an authenticated URL whose valid access time has passed.	None
Request length exceeds defined buffer size	The incoming request is larger than the buffer for the Security Enforcer parser. When the system receives a request that triggers this violation, it stops validating the request for other violations.	None

Table A.2 *Access violations (Continued)*

Length violations

Length violations occur when an HTTP request contains an entity that exceeds the length setting that is defined in the security policy. Table A.3 lists the length violations, describes the event that triggers the violation, and specifies the attack type. Note that all length violations constitute buffer overflow attacks.

Length violation	Violation trigger event	Attack type
Illegal cookie length	The incoming request includes a cookie header that exceeds the acceptable length as specified in the security policy.	Buffer overflow
Illegal header length	The incoming request includes an HTTP header that exceeds the acceptable length as specified in the security policy.	Buffer overflow
Illegal POST data length	The incoming request contains POST data whose length exceeds the acceptable length as specified in the security policy.	Buffer overflow
Illegal query string length	The incoming request contains a query string whose length exceeds the acceptable length as specified in the security policy.	Buffer overflow

Table A.3 *Length violations*

Length violation	Violation trigger event	Attack type
Illegal request length	The incoming request length exceeds the acceptable length as specified in the security policy.	Buffer overflow
Illegal URL length	The incoming request references a URL whose length exceeds the acceptable length as specified in the security policy.	Buffer overflow

Table A.3 *Length violations (Continued)*

Input violations

Input violations occur when an HTTP request includes a parameter or header that contains data or information that does not match, or comply with, the security policy. Input violations most often occur when the security policy contains defined user-input parameters.

Table A.4 lists the input violations, describes the event that triggers the violation, and specifies the attack type (if one is associated with the violation).

Input violation	Violation trigger event	Attack type
Failed to convert character	The incoming request contains a character that does not comply with the encoding of the web application (the character set of the security policy), and the Security Enforcer cannot convert the character to the current encoding.	None
Illegal attachment in SOAP message	The incoming request contains a SOAP message in which there is an attachment that is not permitted by the security policy.	Injection attempt
Illegal dynamic parameter value	The incoming request contains a dynamic parameter whose value was changed illegally on the client side.	Parameter tampering
Illegal empty parameter value	The incoming request contains a parameter whose value is empty when it must contain a value.	None
Illegal meta character in header	The incoming request includes a header whose value contains a meta character that is not allowed in the security policy. Note that if you accept the meta character that caused the violation, the Application Security Manager updates the character set for header values to allow the meta character.	None
Illegal meta character in parameter value	The incoming request includes a parameter whose value contains a meta character that is not allowed in the security policy. Note that if you accept the meta character that caused the violation, the Application Security Manager updates the character set for parameter values to allow the meta character.	None
Illegal number of mandatory parameters	The incoming request contains either too few or too many mandatory parameters on a flow. Note that only flows can contain mandatory parameters.	None
Illegal parameter	The incoming request contains a parameter that is not defined in the security policy.	None

Table A.4 *Input violations*

Input violation	Violation trigger event	Attack type
Illegal parameter data type	The incoming request contains a parameter for which the data type does not match the data type that is defined in the security policy. This violation applies to user-input parameters, which may be defined in the security policy as either integer, alpha-numeric, decimal, phone, or email.	Parameter tampering
Illegal parameter numeric value	The incoming request contains a parameter whose value is not in the range of decimal or integer values defined in the security policy.	Parameter tampering
Illegal parameter value length	The incoming request contains a parameter whose value length does not match the value length that is defined in the security policy. Note that this violation is relevant only for user input parameters.	None
Illegal query string or POST data	The incoming request contains a query string or POST data that is not allowed in a flow.	None
Illegal repeated parameter name	The request contains multiple parameters with the same name, and may indicate an HTTP parameter pollution attack. If this behavior is permitted, you can allow repeated occurrences when creating parameters.	Detection evasion
Illegal static parameter value	The incoming request contains a static parameter whose value is not defined in the security policy.	Parameter tampering
Malformed XML data	The incoming request contains XML data that is not well-formed, according to W3C standards.	XML parser attack
Maximum login attempts are exceeded	Application Security Manager detected too many failed login attempts.	Brute force attack
Null in multi-part parameter value	The incoming multi-part request has a parameter that contains a binary NULL (**0x00**) value and the content-type header parameter type is binary when the parameter is defined in the security policy as user-input alpha-numeric.	None
Parameter value does not comply with regular expression	The incoming request contains an alphanumeric parameter value that does not match the expected pattern specified by the regular-expression field for that parameter.	Parameter tampering
SOAP method not allowed	The incoming request contains a SOAP method that is not permitted by the security policy.	Information leakage
Web scraping detected	The incoming request looks like it is from a non-human, automated source, or illegal web robot.	Web scraping

Table A.4 *Input violations (Continued)*

Input violation	Violation trigger event	Attack type
Web Services Security failure	The request contains one of the following web services security errors: • Internal Error • Malformed Error • Certificate Expired • Certificate Error • Decryption Error • Signing Error • Verification Error • Missing Timestamp • Invalid Timestamp • Expired Timestamp • Timestamp expiration is too far in the future • Unsigned Timestamp	None
XML data does not comply with format settings	The incoming request contains XML data that does not comply with the defense configuration in the XML profile.	XML parser attack
XML data does not comply with schema or WSDL document	The incoming request contains XML data that does not match the schema file or WSDL document that is part of the XML profile.	None

Table A.4 *Input violations (Continued)*

Note

The Application Security Manager does not distinguish between dynamic parameters that are defined incorrectly, and dynamic parameters that actually contain bad values. In both cases, the system issues the ***Illegal parameter*** *violation. You can evaluate the request on the Requests List to determine what caused the violation (see* ***Reporting*** *>>* ***Requests****).*

Cookie violations

Cookie violations occur when the cookie values in the HTTP request do not comply with the security policy. Cookie violations may indicate malicious attempts to hijack private information. Table A.5 lists the cookie violations and describes the event that triggers the violation. None of the cookie violations is associated with an attack type.

Cookie violation	Violation trigger event	Attack type
ASM cookie hijacking (also called Wrong message key)	The incoming request contains an Application Security Manager cookie that was created in another session.	None
Expired timestamp	The time stamp in the HTTP cookie is old, which indicates either the malicious reuse of an outdated cookie, or that a client has been idle for too long, or.	None
Modified ASM cookie	The incoming request contains an Application Security Manager cookie that has been modified or tampered with.	None
Modified domain cookie(s)	The domain cookies in the HTTP request do not match the original domain cookies, or are not defined as allowed modified domain cookies in the security policy.	None

Table A.5 *Cookie violations*

Negative security violations

Negative security violations occur when an incoming request contains a string pattern that matches an attack signature in one of the security policy's attack signature sets, or when a response contains exposed user data, for example, a credit card number.

◆ Note

For more information on attack signatures for security policies, see ***Working with attack signature sets****, on page 11-13.*

Table A.6 lists the negative security violations, describes the event that triggers the violation, and specifies the attack type (if one is associated with the violation).

Negative security violation	Violation trigger event	Attack type
Information leakage detected	The response contains sensitive user data. The Data Guard™ feature determines what data is considered sensitive (for details, see *Masking sensitive data*, on page 6-35).	Information leakage
Virus detected	The request includes a file containing a virus or worm.	Virus detected
Attack signature detected	The incoming request, or the response, contains a pattern that matches an attack signature. ***Note****: The* ***Attack signature detected*** *violation does not appear on the Requests screen for signatures that are in staging.*	Attack type depends on which attack signature triggered the violation

Table A.6 *Negative security violations*

Determining the type of attack detected by an attack signature

If you see an **Attack signature detected** violation associated with a request, you can determine the type of attack that caused the violation.

To determine the type of attack triggered by an attack signature

1. In the navigation pane, expand **Application Security**, point to **Reporting**, then click **Requests**.
   The Requests screen opens.
2. In the editing context area, ensure that the current edited security policy is the one for which you want to examine attack signature violations.
3. In the Requests List, click the illegal request.
   The screen displays the violations associated with the illegal request.
4. If one of the violations listed is **Attack signature detected**, click that violation hyperlink.
   A popup screen opens and shows the name of the attack signature that caused the violation.
5. In the popup screen, click the signature name.
   A new screen opens and shows details about the attack signature, including the attack type, with links to additional documentation.

Filtering requests by attack type

On the Requests screen, you can filter the display of requests by attack type. An ***attack type*** is a category of security breach.

Refer to *Types of attacks that attack signatures detect*, on page 11-3, for descriptions of each attack type.

To filter requests by attack type

1. In the navigation pane, expand **Application Security** and click **Reporting**.
   The Requests screen opens.
2. If the filter options are not visible, on the left of the **Filter** list, click the Show/Hide Filter button ( ).
   The screen displays the custom filter options.
3. For the **Attack Type**, select the type of attack by which to filter the list of requests.
4. Click **Go**.
   If you have illegal requests of that attack type, the screen displays them in the Requests List.

B

Working with the Application-Ready Security Policies

- Understanding application-ready security policies
- Using the Rapid Deployment security policy
- Using the ActiveSync security policy
- Using the OWA Exchange 2003 security policy
- Using the OWA Exchange 2007 security policy
- Using the SharePoint 2003 security policy
- Using the SharePoint 2007 security policy
- Using the Lotus Domino 6.5 security policy
- Using the Oracle Applications 10g security policy
- Using the Oracle Applications 11i security policy
- Using the PeopleSoft Portal 9 security policy
- Using the SAP NetWeaver security policy
- Using the WhiteHat Sentinel Baseline security policy
- Managing large file uploads when using the application-ready security policies

Understanding application-ready security policies

The Application Security Manager™ provides application-ready security policies that are preconfigured to address the security needs of specific enterprise applications. Application security templates create working security policies that can immediately increase the security of an application.

When you select an application-ready security policy, the system automatically populates the security policy with the entities and optimizations that are specific to the application. Application-ready security policies are available to create policies for web applications that use either the HTTP or the HTTPS protocol.

◆ Note

For information on security policies in general, refer to Chapter 6, ***Manually Configuring Security Policies****.*

Using the Deployment wizard to implement application-ready security policies

The Deployment wizard offers a quick and automated method for deploying a security policy for well-known enterprise applications. From the Deployment wizard, you select the manual deployment scenario, then choose the application-ready security policy for the application you want to protect. For more information on working with the Deployment wizard, refer to the ***BIG-IP® Application Security Manager™: Getting Started Guide***.

When you use one of the application-ready security policies, the system builds the security policy in Transparent mode to allow you to review and fine-tune the security policy before it is enforced. After you see that the security policy does not produce any false positives, you can place the security policy in Blocking mode.

You also have the option of starting automated policy building, and having the Policy Builder add to the security policy based on examining the traffic. If you do, the security policy remains in Transparent mode until you set it to blocking. Refer to *Stopping and starting automatic policy building*, on page 5-23 for details on how to start the Policy Builder. For information on how to change the enforcement mode to blocking, see *Configuring the enforcement mode*, on page 6-3.

Using the Rapid Deployment security policy

The Rapid Deployment security policy is configured with a general set of security checks to minimize or eliminate the amount of false-positives, and reduce the complexity and length of the initial evaluation deployment period. By default, the Rapid Deployment security policy is in a globally transparent mode. You can enable blocking either globally or for individual security checks, as necessary. The Rapid Deployment security policy enables organizations to meet the majority of web application security requirements as outlined in PCI DSS v1.2 section 6, FISMA, HIPPA, and others.

Overview of the Rapid Deployment security policy features

When you use the Rapid Deployment security policy to create your security policy, the Application Security Manager automatically configures the following security optimizations:

- Protection against known vulnerabilities, cross-site scripting attacks, and SQL and OS injection attacks
- Evasion attacks detection
- HTTP protocol and HTTP cookie compliance enforcement
- Protection against data leakage in responses, for US Social Security Numbers, credit card numbers, and custom patterns
- Protection against application buffer overflow attacks
- Protection against improper error handling by the application
- Prevention from OS and web server fingerprinting

Using the ActiveSync security policy

The ActiveSync application-ready security policies protect servers running Microsoft® ActiveSync® software, versions 1.0 or 2.0. Templates are available for both the HTTP and the HTTPS protocols.

ActiveSync is Microsoft's protocol to synchronize mobile devices with the corporate Microsoft Exchange Server. Windows mobile and iPhone® devices use ActiveSync to synchronize email, contacts, and calendar data.

Overview of the ActiveSync security policy features

When you use an ActiveSync security policy to create your security policy, the Application Security Manager automatically configures the optimal security policy to protect the ActiveSync application. It also configures attack signatures to detect application-specific attack patterns.

Configuring the system to secure the ActiveSync application

If you are using the ActiveSync security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the ActiveSync application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **ActiveSync v1.0 v2.0** (**http** or **https**) security policy.

◆ Note

*If you are using OWA Exchange 2003 or 2007 with ActiveSync, select the **OWA Exchange 2003/2007 with ActiveSync** security policy.*

Using the OWA Exchange 2003 security policy

The OWA Exchange 2003 application-ready security policies protect servers running Microsoft® Outlook® Web Access (OWA) software with Microsoft® Exchange Server 2003 software. The templates are available for both the HTTP and the HTTPS protocols.

◆ Note

If you are creating a security policy for servers running Microsoft Exchange Server 2007 software, you should use the OWA Exchange 2007 security policy instead of this template. Refer to ***Using the OWA Exchange 2007 security policy****, on page B-5, for more information.*

Overview of the OWA Exchange 2003 security policy features

When you use an OWA Exchange 2003 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the Outlook Web Access application:

- The cookie signing feature prevents session hijacking attacks.
- The Allowed Methods list includes POST, GET, HEAD, OPTIONS, and other methods used by the OWA application.
- Attack signatures detect application-specific attack patterns, including a customized signature that detects attack patterns in Microsoft Internet Explorer® requests.
- General XML security checks run on the OWA application traffic.
- Length violations prevent buffer overflow attacks.

Configuring the system to secure the OWA 2003 application

If you are using an OWA Exchange 2003 security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the OWA application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Configure the web application language.
  - Select the **OWA Exchange 2003** (**http** or **https**) security policy.

◆ Note

If you are using OWA Exchange 2003 with ActiveSync, select the ***OWA Exchange 2003 with ActiveSync*** *security policy.*

Using the OWA Exchange 2007 security policy

The OWA Exchange 2007 application-ready security policies protect servers running Microsoft® Outlook® Web Access (OWA) software with Microsoft® Exchange Server 2007 software. Templates are available for both the HTTP and the HTTPS protocols.

◆ Note

If you are creating a security policy for servers running Microsoft Exchange Server 2003 software, then you should use the OWA Exchange 2003 template instead of this template. Refer to ***Using the OWA Exchange 2003 security policy****, on page B-4, for more information.*

Overview of the OWA Exchange 2007 security policy features

When you use an OWA Exchange 2007 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the Outlook Web Access application:

- The cookie signing feature prevents session hijacking attacks.
- The Allowed Methods list includes POST, GET, HEAD, and OPTIONS.
- Attack signatures detect application-specific attack patterns, including a customized factory signature that detects attack patterns in Internet Explorer requests.
- An XML security profile validates the XML traffic.
- Length violations prevent buffer overflow attacks.

Configuring the system to secure the OWA 2007 application

If you are using an OWA Exchange 2007 security policy, there are several tasks you perform before you create the actual security policy with the template. The tasks are:

- Create a local traffic pool for the application resources.
- Create an application security class for the OWA application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Configure the web application language.
  - Select the **OWA Exchange 2007** (**http** or **https**) security policy.
- Replace the generic domain name in several URLs with your application's domain name.

◆ Note

If using OWA Exchange 2007 with ActiveSync, select the ***OWA Exchange 2007 with ActiveSync*** *security policy.*

Using the SharePoint 2003 security policy

The SharePoint 2003 application-ready security policies protect servers running Microsoft® SharePoint® 2003 software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the SharePoint 2003 security policy features

When you use a SharePoint 2003 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the SharePoint application:

- The Allowed Methods list includes POST, GET, HEAD, and OPTIONS.
- Attack signatures detect SharePoint-specific and generic web-application attack patterns in requests.
- The illegal session ID in URL mechanism removes session ID information to prevent false-positive alarms for the **Illegal URL** violation.

Configuring the system to secure the SharePoint 2003 application

If you are using the SharePoint 2003 security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the SharePoint application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **SharePoint 2003** (**http** or **https**) security policy.

Using the SharePoint 2007 security policy

The SharePoint 2007 application-ready security policies protect servers running Microsoft® SharePoint® 2007 software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the SharePoint 2007 security policy features

When you use a SharePoint 2007 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the SharePoint application:

- The Allowed Methods list includes POST, GET, HEAD, and OPTIONS.
- Attack signatures detect SharePoint-specific and generic web-application attack patterns in requests.

Configuring the system to secure the SharePoint 2007 application

If you are using the SharePoint 2007 security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the SharePoint application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **SharePoint 2007** (**http** or **https**) security policy.

Using the Lotus Domino 6.5 security policy

The Lotus Domino 6.5 application-ready security policies protect servers running Lotus® Domino® software version 6.5.4. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the Lotus Domino 6.5 security policy features

When you use a Lotus Domino 6.5 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the Lotus Domino 6.5 application:

- The cookie signing feature prevents session hijacking attacks.
- Parameter validation prevents parameter tampering attacks.
- Attack signatures detect application-specific attack patterns.
- The illegal session ID in URL mechanism removes session ID information to prevent false-positive alarms for the **Illegal URL** violation.

Configuring the system to protect the Lotus Domino 6.5 application

If you are using the Lotus Domino 6.5 security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the Lotus Domino 6.5 application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Configure the web application language,
  - Select the **Lotus Domino 6.5** (**http** or **https**) security policy.

Using the Oracle Applications 10g security policy

The Oracle Applications 10g application-ready security policies protect servers running the Oracle® Applications 10g database software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the Oracle Applications 10g security policy features

When you use the Oracle Applications 10g security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the Oracle database application:

- The cookie signing feature prevents session hijacking attacks.
- Parameter validation prevents parameter tampering attacks.
- Attack signatures detect application-specific attack patterns.
- Meta characters enforcement detects user-input manipulations.

Configuring the system to protect the Oracle Applications 10g application

If you are using the Oracle Applications 11i security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the Oracle application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **Oracle Applications 10g** (**http** or **https**) security policy.

Using the Oracle Applications 11i security policy

The Oracle Applications 11i application-ready security policies protect servers running the Oracle® Applications 11i database software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the Oracle Applications 11i security policy features

When you use the Oracle Applications 11i security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the Oracle database application:

- The cookie signing feature prevents session hijacking attacks
- Parameter validation prevents parameter tampering attacks
- Attack signatures detect application-specific attack patterns
- Meta characters enforcement detects user-input manipulations

Configuring the system to protect the Oracle Applications 11i application

If you are using the Oracle Applications 11i security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the Oracle application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **Oracle Applications 11i** (**http** or **https**) security policy.

Using the PeopleSoft Portal 9 security policy

The PeopleSoft Portal 9 application-ready security policies protect servers running the PeopleSoft Portal 9 database software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the PeopleSoft Portal 9 security policy features

When you use the PeopleSoft Portal 9 security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the database application:

- The cookie signing feature prevents session hijacking attacks.
- Parameter validation prevents parameter tampering attacks.
- Attack signatures detect application-specific attack patterns.
- Meta characters enforcement detects user-input manipulations.

Configuring the system to protect the PeopleSoft Portal 9 application

If you are using the PeopleSoft Portal 9 security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the Oracle application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **PeopleSoft Portal 9** (**http** or **https**) security policy.

Using the SAP NetWeaver security policy

The SAP NetWeaver application-ready security policies protect servers running the SAP NetWeaver® 7 software. The templates are available for both the HTTP and the HTTPS protocols.

Overview of the SAP NetWeaver security policy features

When you use an SAP NetWeaver security policy to create your security policy, the Application Security Manager automatically configures the following optimizations to protect the SAP NetWeaver application:

- The cookie signing feature prevents session hijacking attacks.
- Parameter validation prevents parameter tampering attacks.
- Attack signatures detect application-specific attack patterns.
- Meta characters enforcement detects user-input manipulations.

Configuring the system to protect the SAP NetWeaver application

If you are using the SAP NetWeaver security policy, you must perform the following tasks to create the security policy with the template:

- Create a local traffic pool for the application resources.
- Create an application security class for the SAP NetWeaver application.
- Create a local traffic virtual server.
- Run the Deployment wizard:
  - Set the web application language.
  - Select the **SAP NetWeaver 7** (**http** or **https**) security policy.

Using the WhiteHat Sentinel Baseline security policy

You can select the WhiteHat Sentinel Baseline application-ready security policy if deploying using the WhiteHat Sentinel Scanner software. WhiteHat Sentinel, integrated with Application Security Manager, provides scanning technology that identifies, manages, and remediates website vulnerabilities.

Overview of the WhiteHat Baseline security policy features

When you use the WhiteHat Sentinel Baseline security policy to create your security policy, the Application Security Manager automatically configures a baseline security policy to work with WhiteHat Sentinel. Through integration with Application Security Manager, the WhiteHat Sentinel service can configure security policy rules to protect against vulnerabilities discovered in a web application.

With the WhiteHat Sentinel Baseline security policy, you can protect applications against cross-site scripting, SQL injection, predictable resource location, command injection, XPath injection, path traversal, and HTTP response splitting.

Configuring the system to work with WhiteHat Sentinel

If you are using the WhiteHat Sentinel Baseline security policy, you must perform the following tasks to create the security policy:

- Create a local traffic pool for the application resources.
- Create an application security class for the application you want to protect.
- Create a local traffic virtual server.
- Run the Application Security Manager Deployment wizard:
  - Set the web application language.
  - Select the **WhiteHat Sentinel Baseline** security policy.
- In the WhiteHat Sentinel user interface, run the scanner from the Schedule screen.
- In the WhiteHat Sentinel user interface, review the vulnerabilities on the Details screen and update the Application Security Manager security policy to mitigate the found vulnerabilities.

If you have an existing security policy protecting a web application, in WhiteHat Sentinel, when you select the web application in the firewall area, a WhiteHat-baseline security policy is automatically created. F5 Networks recommends that you use the WhiteHat-baseline security policy to protect the application.

Managing large file uploads when using the application-ready security policies

The web applications for which you can use one of the application-ready security policies to configure a security policy frequently experience large file uploads (larger than 10 MB files). As a result, you may encounter clients that are blocked due to the large file uploads, and should not be. You can resolve this issue by disabling the Block flag for the security policy violation, **Request length exceeds defined buffer size**. By disabling the blocking action for this violation, the Security Enforcer inspects the headers in the associated request, but ignores the file upload itself.

◆ Note

For more information on the blocking policy and the enforcement modes, refer to ***Configuring security policy blocking****, on page 6-41.*

To disable the Block flag for the Request length exceeds defined buffer size violation

1. In the navigation pane, expand **Application Security** and click **Policy**.
   The Policy Properties screen opens.
2. From the Blocking menu, choose Settings.
   The Blocking Policy screen opens.
3. In the editing context area, ensure that the edited security policy is the one you want to update.
4. In the Configuration area, ensure that the **Enforcement Mode** setting has the **Blocking** option enabled.

   Note*: You can change the Block flags only when the enforcement mode is* ***Blocking****.*
5. In the Access Violations area, locate the **Request length exceeds defined buffer size** violation, and in the Block column, clear the Block check box.
6. Click the **Save** button to save any changes you may have made on this screen.
7. To put the security policy changes into effect immediately, click the **Apply Policy** button in the editing context area.

C

Syntax for Creating User-Defined Attack Signatures

- Writing rules for user-defined attack signatures
- Overview of rule option scopes
- Syntax for attack signature rules

Writing rules for user-defined attack signatures

Attack signatures are composed of several different rule options and modifiers that you can combine to form complex rules that define the exact characteristics of the input to be matched. This appendix describes the syntax and usage for the rule options and modifiers that you need to follow when writing attack signatures. For information on attack signatures in general, refer to Chapter 11, *Working with Attack Signatures*.

Understanding the rule options

The individual unit or rule building block is called a ***rule option***. You can combine the various rule options into a single rule, with an implied AND operator between them. This means that all rule options must be satisfied for the rule to match. For more information on combining rule options, see *Syntax considerations for parameter attack signatures*, on page C-14.

Using the keyword rule options

The keyword rule options search for specific fixed strings in different parts of the input, which are referred to as scopes. (See *Overview of rule option scopes*, on page C-3, for more information on scopes.) Table C.1 lists the keyword rule options and their general usage.

Keyword	Usage
content	Match in the full content. See *Using the content rule option*, on page C-5, for syntax information.
uricontent	Match in the URI, including the query string (unless using the **objonly** modifier). See *Using the uricontent rule option*, on page C-5, for syntax information.
headercontent	Match in the HTTP headers. See *Using the headercontent rule option*, on page C-6, for syntax information.
valuecontent	Matches an alpha-numeric user-input parameter (or an extra-normalized parameter, if using the **norm** modifier); used for parameter values and XML objects. See *Using the valuecontent rule option*, on page C-6, for syntax information, and *Scope modifiers for the pcre rule option*, on page C-3, for more information on scope modifiers. An XML payload is checked for attack signatures when the **valuecontent** keyword is used in the signature. ***Note:*** *The* ***valuecontent*** *parameter replaces the* ***paramcontent*** *parameter that was used in the Application Security Manager™ versions earlier than 10.0.*
reference	Provides an external link to documentation and other information for the rule. See *Using the reference rule option*, on page C-8, for syntax information.

Table C.1 *Attack signature keywords and usage*

Using the pcre rule option

The **pcre** rule option performs a regular expression match on different parts of the input, and is based on the Perl-compatible regular expressions (PCRE) syntax.

Using keyword modifiers for rule options

The keyword modifiers alter the meaning of the rule options. Table C.2 lists the keyword modifiers and their usage.

Keyword modifier	Usage
nocase	The preceding keyword is not case sensitive. See *Using the nocase modifier*, on page C-8, for syntax information.
offset	The preceding keyword is found not less than X bytes into the appropriate scope. This is an absolute modifier. See *Using the offset modifier*, on page C-9, for syntax information.
depth	The preceding keyword is found not more than X bytes into the appropriate scope. This is an absolute modifier. See *Using the depth modifier*, on page C-9, for syntax information.
distance	The immediately preceding keyword is found not less than X bytes after the prior keyword. This is a relative modifier. See *Using the distance modifier*, on page C-10, for syntax information.
within	The immediately preceding keyword is found not more than X bytes after the prior keyword. This is a relative modifier. See *Using the within modifier*, on page C-11, for syntax information.
objonly	Limit the scope of the preceding **uricontent** keyword to the URI part only. See *Using the objonly modifier*, on page C-12, for syntax information.
norm	Matches on the preceding parameter to which additional normalizations have been applied. See *Using the norm modifier*, on page C-12, for syntax information.
xmlonly	Matches on XML objects when used with the **valuecontent** keyword modifier. Refer to *Scope modifiers for the pcre rule option*, on page C-3, for more information.
httponly	Matches on parameters when used with the **valuecontent** keyword modifier. Refer to *Scope modifiers for the pcre rule option*, on page C-3,

Table C.2 *Rule option modifiers*

Using the not character (!) with keyword and pcre rule options

You can use the optional not character (**!**) before the keyword and **pcre** rule options. This specifies that the rule is only matched if the specified option is not matched. Refer to *Syntax for attack signature rules*, on page C-5, for more details on the use of this modifier.

Overview of rule option scopes

Scopes are the elements of a request or a response to which the rule option applies. Most of the rule options apply to request elements, however, some can apply to response bodies. Table C.3 lists the scopes and their corresponding rule options to use in the attack signature.

Scope	Corresponding rule option
Full content of the request, also the response body	Use the **content** keyword. For additional information, see *Using the content rule option*, on page C-5.
URI, including query string	Use the **uricontent** keyword. For additional information, see *Using the uricontent rule option*, on page C-5.
URL only (URI without query string)	Use the **uricontent** keyword with **objonly** modifier. For additional information, see *Using the headercontent rule option*, on page C-6, and *Using the objonly modifier*, on page C-12.
HTTP headers	Use the **headercontent** keyword. For additional information, see *Using the headercontent rule option*, on page C-6.
HTTP parameters in query string or POST data	Use the **valuecontent** keyword. For additional information, see *Using the valuecontent rule option*, on page C-6.
HTTP parameters with additional normalizations	Use the **valuecontent** keyword with the **norm** modifier. For additional information, see *Using the valuecontent rule option*, on page C-6, and *Using the norm modifier*, on page C-12.

Table C.3 *Request scopes and rule options*

Scope modifiers for the pcre rule option

You can modify the **pcre** rule option to apply to any of the scopes described in the previous section, *Overview of rule option scopes*. For the **pcre** rule option, you can use the modifiers described in Table C.4.

PCRE modifiers	Description
None	If you do not specify a modifier, the **pcre** rule option applies to either the full content of the request, or the response body.
U	The **U** modifier specifies the URI scope.
O	The **O** modifier specifies the URL only scope.
H	The **H** modifier specifies the HTTP headers scope.

Table C.4 *Description of **pcre** modifiers*

PCRE modifiers	Description
P	The **P** modifier specifies the parameters scope.
N	The **N** modifier specifies the parameters with additional normalizations scope.
V	The **V** modifier specifies the combined parameters scope and normalization scope.

***Table C.4** Description of **pcre** modifiers (Continued)*

A note about normalization

For the URI and parameter scopes, the system always applies a normalization process before applying the rule options. For parameters, you can also specify the **norm** modifier with the **valuecontent** keyword to have the system perform additional normalizations. The additional parameter normalizations help mitigate common evasion techniques used in XSS, SQL-Injection and Command Execution attacks.

◆ Note

*Applying the **norm** modifier to the **valuecontent** keyword may boost the effectiveness of certain signatures, which, in turn, may cause an increased number of false-positives.*

Syntax for attack signature rules

The following sections describe the usage and provide syntax examples for each of the rule options and modifiers. When you write a rule, use the semicolon character to separate the rule options, and use the colon character to separate option keywords and their arguments. Note that arguments which are strings must be in quotation marks.

Using the content rule option

The **content** rule option matches when the specified string is found anywhere in the full content of the request. The string match is case-sensitive, and must be exact. You can use the not character (!) in front of the string if you want the rule to match when it does not find the exact specified string. Figure C.1 shows syntax examples for the **content** keyword.

```
content:"ABC";
content:!"ABC";
```

Figure C.1 *Syntax examples for the* ***content*** *keyword*

You can use the **content** keyword for request or response attack signatures. If you want the attack signature to apply to responses, there are two additional actions:

- Ensure that you check the **Check Response** setting for the related file type.
- In the rule itself, set the **Apply to** option to **Response**.

◆ Note

The system does not perform any normalizations for the ***content*** *rule option.*

Using the uricontent rule option

The **uricontent** rule option matches when the specified string is found anywhere in the normalized URI, including the query string. The string match is case-sensitive, and must be exact. You can use the not character (!) in front of the string if you want the system to match when it does not find the exact specified string. Figure C.2 shows syntax examples for the **uricontent** keyword.

```
uricontent:"ABC";
uricontent:!"ABC";
```

Figure C.2 *Syntax examples for the* ***uricontent*** *keyword*

You can use the **uricontent** keyword for request attack signatures only.

Using the headercontent rule option

The **headercontent** rule option matches when the specified string is found anywhere in the HTTP request headers. The string match is case-sensitive, and must be exact. You can use the not character (!) in front of the string if you want the rule to match when it does not find the exact specified string. Figure C.3 shows syntax examples for the **headercontent** keyword.

```
headercontent:"ABC";
headercontent:!"ABC";
```

Figure C.3 *Syntax examples for the* ***headercontent*** *keyword*

You can use the **headercontent** keyword for request attack signatures only.

Note

The system does not perform any normalizations for the ***headercontent*** *rule option.*

Using the valuecontent rule option

The **valuecontent** rule option matches when the specified string is found anywhere within a specific alpha-numeric user-input parameter. The system applies **valuecontent** rules only to parameters defined in the security policy. The rule matches against each parameter separately, on the full name and value pair. The string match is case-sensitive, and must be exact. You can use the not character (!) in front of the string if you want the rule to match when it does not find the exact specified string. Figure C.4 shows syntax examples for the **valuecontent** keyword.

```
valuecontent:"ABC";
valuecontent:!"ABC";
```

Figure C.4 *Syntax examples for the* ***valuecontent*** *keyword*

You can use the **valuecontent** keyword for request attack signatures only.

Note

You cannot combine this scope with any other scopes in a single rule.

Using the pcre rule option

The **pcre** rule option matches if the regular expression found within the slash (/) characters matches. The scope of the match depends on the **pcre** modifiers that you specify. For details about the syntax used within the regular expression itself, refer to the **pcre** documentation, which is available

at **http://pcre.org**. For details on the **pcre** modifiers, refer to *Summary of pcre modifiers*, following. Figure C.5 shows syntax examples for the **pcre** keyword.

```
pcre:"/<regex>/";
pcre:"/<regex>/<options>";
pcre:!"/<regex>/";
```

Figure C.5 *Syntax examples for the* ***pcre*** *rule option*

Summary of pcre modifiers

You can use the following modifiers with the **pcre** rule option. Table C.5 describes the scope modifiers.You can use only one scope modifier for the **pcre** rule option.

Scope modifier	Restricts match to scope
None	Full content
U	URI
O	URL
H	Headers
P	Parameter
N	Normalized parameter
V	Parameter and value pairs or XML payloads

Table C.5 *Scope modifiers for the* ***pcre*** *rule option*

Table C.6 describes the matching action modifiers. You can use one or more matching action modifier.

Matching action modifier	Effect
i	The match is not case-sensitive.
s	Change the dot character (**.**) to match any character whatsoever, including a new line, which normally it would not match.

Table C.6 *Matching action modifiers for* ***pcre*** *rule option*

Matching action modifier	Effect
m	Change the caret character (**^**) and the dollar sign character (**$**) from matching the start or end of the scope to matching the start or end of any line anywhere within the scope.
R	The match is relative to the end of the last keyword match. (This modifier is similar to the **distance:0;** modifier.)

***Table C.6** Matching action modifiers for **pcre** rule option (Continued)*

Using the reference rule option

Use the **reference** rule option in a rule to provide an external reference or link to information regarding the rule, its source, or any other relevant documentation. Figure C.6 shows syntax examples for the **reference** keyword.

```
reference:url,www.reference.com;
reference:bugtraq,1234;
reference:cve,2007-1234;
reference:nessus,1234;
```

***Figure C.6** Syntax examples for the **reference** rule option*

Table C.7 lists the reference types.

Type	Value	Example
url	URL	reference:url,www.reference.com;
bugtraq	Bugtraq ID	reference:bugtraq,1234;
cve	CVE ID	reference:cve,2007-1234;
nessus	Nessus Plugin ID	reference:nessus,1234

***Table C.7** Reference types for the **reference** rule option*

Using the nocase modifier

Use the **nocase** modifier with one of the keyword rule options to make it not case-sensitive. Figure C.7 shows syntax examples for the **nocase** modifier.

```
content:"ABC"; nocase;
```

***Figure C.7** Syntax example for the **nocase** modifier*

Using the offset modifier

Use the **offset** modifier to specify that the previous keyword matches within its scope beginning not less than the specified number of bytes from the beginning of the scope. Figure C.8 shows syntax examples for the **offset** modifier.

```
content:"ABC"; offset:10;
uricontent:"ABC"; offset:10;
```

Figure C.8 *Syntax examples for the* ***offset*** *modifier*

For example, the **content** rule in Figure C.8 matches these requests:

```
12345678901234567890
GET /67890ABC ...
GET /678901ABC ...
```

but not these requests:

```
12345678901234567890
GET /6789ABC ...
GET /678ABC ...
```

Tip

The line of numbers above the request examples counts the number of bytes.

You can use the **offset** modifier to modify keywords for any scope. The scope determines where the offset matching begins. For example, the rule **uricontent:"ABC"; offset:10;** matches these requests:

```
xxxx123456789012345
GET /234567890ABC ...
GET /2345678901ABC ...
```

but not these requests:

```
xxxx123456789012345
GET /23456789ABC ...
GET /2345678ABC ...
```

Using the depth modifier

Use the **depth** modifier to specify that the previous keyword matches within its scope ending not more than the specified number of bytes from the beginning of the scope. Figure C.9 shows syntax examples for the **depth** modifier.

```
content:"ABC"; depth:10;
uricontent:"ABC"; depth:10;
```

Figure C.9 *Syntax examples for the* ***depth*** *modifier*

For example, the **content** rule in Figure C.9 matches these requests:

```
12345678901234567890
GET /67ABC ...
GET /6ABC ...
```

but not these requests:

```
12345678901234567890
GET /678ABC ...
GET /6789ABC ...
```

Tip

The line of numbers above the request examples counts the number of bytes.

You can use the **depth** modifier to modify keywords for any scope. The scope determines where the **depth** matching begins. For example, in Figure C.9, the rule **uricontent:"ABC"; depth:10;** matches these requests:

```
xxxx123456789012345
GET /234567ABC ...
GET /23456ABC ...
```

but not these requests:

```
xxxx123456789012345
GET /2345678ABC ...
GET /23456789ABC ...
```

You can combine the **offset** and **depth** modifiers to define both the beginning and ending boundaries of the area in which the keyword can match. For example, the rule **content:"ABC"; offset:10; depth:20;** matches these requests:

```
12345678901234567890123456
GET /67890ABC ...
GET /678901234567ABC ...
```

but not these requests:

```
12345678901234567890123456
GET /678ABC ...
GET /6789ABC ...
GET /67890123456789ABC ...
GET /678901234567890ABC ...
```

Using the distance modifier

Use the **distance** modifier to specify that the previous keyword matches not less than the specified number of bytes from the prior keyword. The **distance** modifier is similar to the **offset** modifier. The **distance** modifier enforces a minimum number of bytes relative to the end of the previously

specified keyword, while the **offset** modifier is an absolute value that starts matching from the beginning of the corresponding keyword scope. Figure C.10 shows a syntax example for the **distance** modifier.

```
content:"ABC"; content:"XYZ"; distance:10;
```

Figure C.10 *Syntax example for the* ***distance*** *modifier*

The example rule shown in Figure C.10 matches these requests:

```
xxxxxxxx12345678901234567890
GET /ABC1234567890XYZ ...
GET /ABC12345678901XYZ ...
```

but not these requests:

```
xxxxxxxx12345678901234567890
GET /ABC123456789XYZ ...
GET /ABC12345678XYZ ...
```

◆ Tip

The line of numbers above the request examples counts the number of bytes.

Use the **distance** modifier when the rule includes two keywords, and you want to enforce that the second keyword appears (anywhere) after the first keyword. Note that without the **distance:0;** modifier, no positional relationship exists between two keywords in a rule. As such, the rule **content:"ABC"; content:"XYZ";**, without the **distance** modifier, matches both of these requests:

```
GET /ABCXYZ ...
GET /XYZABC ...
```

Using the within modifier

Use the **within** modifier to specify that the previous keyword matches not more than the specified number of bytes from the prior keyword. The **within** modifier is similar to the **depth** modifier, except that the **within** modifier enforces a maximum number of bytes relative to the end of the previously specified keyword, while the **depth** modifier is an absolute value that starts matching from the beginning of the corresponding keyword scope. Figure C.11 shows a syntax example for the **within** modifier.

```
content:"ABC"; content:"XYZ"; within:10;
```

Figure C.11 *Syntax example for the* ***within*** *modifier*

For example, the rule in Figure C.11 matches these requests:

```
xxxxxxxx12345678901234567890
GET /ABC1234567XYZ ...
GET /ABC123456XYZ ...
```

but not these requests:

```
xxxxxxxx12345678901234567890
GET /ABC12345678XYZ ...
GET /ABC123456789XYZ ...
```

Tip

The line of numbers above the request examples counts the number of bytes.

You can combine the **distance** and **within** modifiers to define both the beginning and ending boundaries of the area in which the keyword can match, relative to the end of the previous keyword match. For example, the rule **content:"ABC"; content:"XYZ"; distance:10; within:20;** matches these requests:

```
xxxxxxxx12345678901234567890
GET /ABC1234567890XYZ ...
GET /ABC12345678901234567XYZ ...
```

but not these requests:

```
xxxxxxxx123456789012345678901234567890
GET /ABC123456789XYZ ...
GET /ABC123456789012345678XYZ ...
```

Using the objonly modifier

Use the **objonly** modifier with the **uricontent** keyword to specify that the match must be found within the URI and not the query string. For example, in the URI, **GET /index.html?q=1,** the object part is **/index.html**. Figure C.12 shows a syntax example for the **objonly** keyword.

```
uricontent:"ABC"; objonly;
```

Figure C.12 *Syntax example for the* ***objonly*** *modifier*

For example, the rule shown in Figure C.12 matches these requests:

```
GET /ABC ...
GET /ABC?param=123 ...
```

but not this request:

```
GET /123?param=ABC ...
```

Using the norm modifier

Use the **norm** modifier to specify that the system applies additional normalization processes to parameter and value pairs before applying the rule. The additional normalization processes include transformations to mitigate evasion techniques commonly used in cross-site scripting (XSS),

SQL-Injection, and Command Execution attacks. Refer to *A note about normalization*, on page C-4, for more information on normalization. Figure C.13 shows a syntax example for the **norm** modifier.

```
valuecontent:"ABC"; norm;
```

Figure C.13 *Syntax example for the* ***norm*** *modifier*

Note

The ***norm*** *modifier applies only to the* ***valuecontent*** *rule option. See* ***Using the valuecontent rule option****, on page C-6, for additional information.*

Using character escaping

For user-defined attack signature rules, you can use the pipe symbol (|) to escape special characters that cannot easily be represented as-is in the keyword arguments. You use the ASCII-equivalent hexadecimal values to represent the characters in the argument. Figure C.14 shows syntax examples for escaping characters.

```
content:"ABC|00|XYZ";
content:"ABC|22 22|XYZ";
```

Figure C.14 *Syntax examples for escaping characters*

The system escapes all of the values that occur between the two pipe symbols in the argument. For example, the first rule in Figure C.14, where **|00|** represents the null character, matches the string **ABC<NULL>XYZ**. The second rule in Figure C.14, where **|22 22|** represents two double quotation marks, matches the string **ABC""XYZ**.

Use the pipe symbol to escape the following characters when you use them in a keyword argument:

- Colon (**:**)
- Semicolon (**;**)
- Double quotation mark (**"**)
- Backward slash (\)
- Pipe (|)
- All binary characters (not ASCII-printable characters), including:
  - ASCII 0x00 through 0x1F
  - ASCII 0x7F through 0xFF
- F5 Networks recommends that you escape the space character (ASCII 0x20), as well.

Note that for the **pcre** rule option, you use the **\x** escape sequence, and not the pipe symbols, to escape characters. See the PCRE documentation, which is available at **http://pcre.org**, for more information. The list of characters that you must escape is the same as those that apply to the other rule options.

Syntax considerations for parameter attack signatures

Any attack signature that contains the **valuecontent** option in its rule is considered a parameter signature, that is, an attack signature that applies to the user-input, alpha-numeric parameters that are defined within a security policy. The system applies parameter attack signatures to each parameter name and value pair (**name=value**) individually.

You cannot use the **valuecontent** option with other **content** rule options.You can disable the parameter attack signature at both the parameter level, and the security policy level.

Syntax considerations for response attack signatures

Response attack signature rules can contain only rule options and modifiers that apply to the entire content of the response. In other words, for response rules, you can use the **content**, **threshold**, and **reference** keywords, and any applicable modifiers for these keywords. You can also use the **pcre** rule option for responses, as long as you do not use a scope modifier for the **pcre** keyword.

Combining rule options

You can combine rule options together to form composite rules. The rule options (or ***option clusters***, since you can combine several rule options to form a single ***assertion***, by using keywords combined with modifiers) are combined with an implied AND operator, so that all of the conditions specified in a rule must be satisfied in order to satisfy the rule as a whole.

It is important to be aware of the following points when combining rule options:

- You can combine different scopes within one rule as long as you do not use any relative modifiers. For example, the following rule is invalid because it includes two scopes (**content** and **uricontent**), and **within** is a relative modifier.

  ```
  content:"ABC"; uricontent:"XYZ"; within:10;
  ```

- You cannot combine the **valuecontent** rule option, nor the **pcre P** rule option, with other scope keywords. The parameter rule options must be the only scope keywords in their respective rules. You can, however, combine the parameter keywords with additional **valuecontent** or **pcre P** keywords, including those that have the **norm** (or **N**, for **pcre**) modifier.

Rule combination example

It is important that you do not combine rules that conflict. The following examples demonstrate both a valid rule combination and an invalid combination, where there are conflicting or illegal rule options and keywords.

```
signature: valuecontent:"AB23XYZ4"
pcre:      "/list-style-image.*?\:.*?url/Psi";
```

Figure C.15 *Valid combined-rule example for the* ***valuecontent*** *keyword*

Result: OK

```
Signature: valuecontent:"AB23XYZ4";
pcre:      "/list-style-image.*?\:.*?url/Usi";
```

Figure C.16 *Invalid combined-rule example for the* ***valuecontent*** *keyword*

Error message: Invalid rule.

Combination Error: HTTP-based value content and general content cannot be combined in a single rule.

The rule combination in Figure C.16 is invalid because of the **U** modifier. The **U** modifier indicates that the **pcre** expression should match the URI scope of the request. You cannot combine the **U** modifier with the **paramcontent** keyword.

D

Internal Parameters for Advanced Configuration

- Overview of internal parameters
- Viewing internal parameters
- Restoring the default settings for internal parameters

Overview of internal parameters

Several internal parameters control how the Application Security Manager™ functions. In most cases, you do not need to change the internal parameters from their default settings. Table D.1 lists the internal parameters, their default values, and a description of their purpose.

◆ Note

F5 Networks recommends that you change the values of parameters only with the guidance of Technical Support.

Internal Parameter	Default Value	Description
allow_all_cookies_at_entry_point	0 (Boolean value)	Specifies, when set to **0**, that if a request arrives with no main ASM cookie (entry point) then every domain cookie that is not configured as an allowed cookie is considered an illegal domain cookie. When set to **1**, all cookies are accepted at entry points.
cookie_digest_key	1111222233334444555 5666677778888 (key)	Provides a key in the MD5 digest calculations for ASM cookies. ***Note**: For security reasons, F5 Networks recommends that you change the cookie digest key from the default value. When changing the value for the key, use the same key value for units in a redundant pair, by configuring the setting on one system and performing a **ConfigSync** with the redundant pair member.*
cookie_expiration_time_out	600 seconds	Allows the Security Enforcer to determine the time (in seconds) for which the ASM cookie data is valid.
cookie_max_age	0 seconds	Specifies the maximum age value (in seconds) assigned to the Max-Age attribute of the ASM cookie. When set to **0**, ASM cookies never expire.
cookie_renewal_time_stamp	300 seconds	Defines how often the Security Enforcer renews the ASM cookie time. This internal parameter is tightly coupled with **cookie_expiration_time_out** (in seconds).
ecard_max_http_req_uri_len	2048 bytes	Defines a maximum URI length that the Security Enforcer can support in its internal buffers. If this number is higher (more permissive) than the internal URI-length limit defined per file type, the internal file-type limit is the actual limit. Exceeding this internal limit triggers the **HTTP protocol compliance failed** violation.
ecard_regexp_decimal	^\s*[+-]?\d*(\.\d+)?\s*$ (regular expression)	Specifies the regular expression that defines a valid pattern for parameter values of type **decimal**.

***Table D.1** Internal parameters for the Application Security Manager*

Internal Parameter	Default Value	Description
ecard_regexp_email	^\s*([\w.-]+)@([\w.-]+)\s*$ (regular expression)	Specifies the regular expression that defines a valid pattern for parameter values of type **email**.
ecard_regexp_phone	^\s*[0-9 ()+-]+\s*$ (regular expression)	Specifies the regular expression that defines a valid pattern for parameter values of type **phone number**.
LogSignatures	1(Enabled)	Specifies that the system keeps track of attack signatures that have been disabled (either globally or on the parameter level) by accepting learning suggestions. A signature may have been disabled due to a false positive. When set to **0**, the system does not track disabled signatures.
long_request_buffer_size	10000000 bytes	Specifies the longest request length supported by the Security Enforcer.
MaxFtpSessions	5000 sessions	Specifies the maximum number of concurrent FTP connections that the Protocol Security Module can manage.
MaximumCryptographicOperations	32 operations	Specifies the maximum number of cryptographic operations allowed per document by Web Services encryption and decryption.
MaxJobs	15000 sessions	Specifies the maximum number of concurrent sessions that the Security Enforcer can handle.
MaxSmtpSessions	3000 sessions	Specifies the maximum number of concurrent SMTP connections that the Protocol Security Module can manage.
MaxViolationEntries	500 entries	Specifies the maximum number of violation entries per violation type kept in memory. Note that this parameter applies only to the security profiles in the Protocol Security Module.
max_concurrent_long_request	100 requests	Specifies the maximum number of concurrent long requests that the Security Enforcer can handle. A long request is a request longer than **request_buffer_size** and less than **long_request_buffer_size**.
max_filtered_html_length	52428800 bytes	Defines the maximum size of responses retained by the system.
OverviewEnabled	1 (Boolean value)	Specifies, when set to **1**, that data collection is enabled for both the graphs on the Overview screen and also for the Denial of Service attack prevention feature. When set to **0**, data collection is disabled.

Table D.1 *Internal parameters for the Application Security Manager (Continued)*

Internal Parameter	Default Value	Description
ProtocolIndication	-1	Specifies how the system distinguishes between HTTP and HTTPS URLs. If the value is **-1**, the system decides whether the object requested is an HTTP request or an HTTPS request based on the incoming traffic. If the value is **0**, the system treats all incoming URL requests as HTTP requests. If the value is **1**, the system treats all incoming URL requests as HTTPS requests.
PRXRateLimit	200 requests per second	Specifies the number of requests per second that the Security Enforcer can enter into the proxy log.
request_buffer_size	10000 bytes	Specifies the common request length supported by the Security Enforcer.
ResponseBufferSize	131072 bytes	Specifies the maximum buffer size for a single instance of the accumulated response buffers. The system accumulates response buffers until their total size reaches the **max_filtered_html_length**.
RWLightThreads	0 (number of CPU cores determines number of threads)	Specifies, when the value is greater than zero, the number of threads that the Security Enforcer uses for protocol security. When the value is **0**, the number of CPU cores in the system determines the number of threads.
RWThreads	0 (number of CPU cores determines number of threads)	Specifies, when the value is greater than zero, the number of threads that the Security Enforcer uses for application security. When the value is **0**, the number of CPU cores in the system determines the number of threads.
total_umu_max_size	0 KB	Specifies the maximum memory size (in kilobytes) available for the Security Enforcer's memory pools.
total_xml_memory	0 bytes	Specifies the maximum amount of memory that can be allocated to the XML parser. A value of **0** means no limit to the amount of memory that the parser can use.
virus_header_name	X-Virus-Name (McAfee's default response header)	Specifies the header name used by an anti-virus program on an ICAP server. By default, the system supports an ICAP server with McAfee anti-virus protection. If you are using a different ICAP server, change this to the appropriate header value.

Table D.1 *Internal parameters for the Application Security Manager (Continued)*

Viewing internal parameters

You can review the settings for the internal parameters on the Advanced Configuration screen.

To view internal parameters in the Configuration utility

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. On the menu bar, click **Advanced Configuration**.
   The Advanced Configuration screen opens, where you can review the settings for the internal parameters.

Note

F5 Networks recommends that you change the values for the internal parameters only with the guidance of the technical support staff.

Restoring the default settings for internal parameters

If you change any of the parameter values for the internal parameters, it is easy to restore the default settings for those values.

To restore the internal parameter default settings

1. In the navigation pane, expand **Application Security** and click **Options**.
   The Attack Signatures screen opens.
2. On the menu bar, click **Advanced Configuration**.
   The Advanced Configuration screen opens.
3. Above (or below) the Advanced Configuration area, click the **Restore Defaults** button.
   The system resets any changed parameter values to their factory settings.
4. From the command line, run the following command to restart the system:

   ```
   bigstart restart asm
   ```

   The system restarts using the default values for all internal parameters.

E

Upgrading HTTP Security Profiles to Security Policies

- Overview of the Migration wizard
- Performing the migration

Overview of the Migration wizard

Customers who want to upgrade from the BIG-IP® Protocol Security Module™ to BIG-IP® Application Security Manager™ can use the Migration wizard to facilitate the upgrade process. The Migration wizard converts an HTTP security profile in the Protocol Security Module configuration to a security policy for a web application in Application Security Manager. If you are not familiar with the features of Application Security Manager, F5 Networks recommends you review this configuration guide before you perform the migration.

The Migration wizard is available only after you have licensed the BIG-IP Application Security Manager module.

Important

You cannot reverse the migration process after converting Protocol Security Module security profiles into security policies in Application Security Manager.

Performing the migration

The Migration wizard guides you through the steps necessary to convert HTTP security profiles in Protocol Security Module to baseline security policies in Application Security Manager. As part of the migration, you create an application security class to replace the HTTP security profile. For detailed information on application security classes, refer to Chapter 3, *Working with Application Security Classes*.

To perform the migration

1. In the navigation pane, expand **Protocol Security** and click **Migration**.
   The Virtual Servers Selection screen of the Migration wizard opens. The wizard automatically detects the virtual servers whose HTTP traffic profiles are associated with HTTP security profiles.
2. For the **Virtual Server** setting, select the virtual server for which you want to create an application security class.
3. For the **Class** setting, select the appropriate option to indicate the class you want to use:
   - **Create new**
     Then, in the box, type a name for the new application security class, using only alphanumeric characters and the underscore character.
   - **Use existing**
     Then, select an existing application security class from the list.
4. Click **Next**.
   The Finish screen opens.
5. Complete the migration process as appropriate:
   - If you created a new application security class, the migration completes after you use the Deployment wizard to configure the security policy. See Chapter 5, *Building a Security Policy Automatically*, for more information.
   - If you used an existing application security class, click the **Finish** button on the Web Application Properties screen to complete the migration.

◆ Note

If you apply a security policy application template, the template overrides any settings that may have been imported by the Migration wizard.

F

Running Application Security Manager on the VIPRION Chassis

- Overview of running Application Security Manager on the VIPRION chassis
- Viewing VIPRION cluster member synchronization status

Overview of running Application Security Manager on the VIPRION chassis

In contrast to the way the Application Security Manager™ runs in a redundant system configuration, where only the active unit handles requests and enforcement, on the VIPRION® system, the primary and secondary cluster members handle traffic and enforcement. A separate instance of the Application Security Manager runs on each of the cluster members in the VIPRION system. In the event of blade failure in the chassis, updates and synchronization gracefully and transparently transfer security policies and data to the new primary cluster member.

The Application Security Manager system failover communication on the VIPRION chassis is the same as that in redundant system configurations, ensuring that configuration data are synchronized to all cluster members in the cluster. Policy Builder and Learning Manager run only on the primary member. When configuration or security policy changes are made to the cluster, the active security policy is copied from the primary member to those that are designated as secondary cluster members. Each secondary cluster member imports the updated security policy and sets it to the active state.

The Application Security Manager functionality is the same on the VIPRION chassis as it is when installed on a single cluster member or as a standalone component, with the following exceptions:

- You can view the availability status of Application Security Manager on the High Availability screen.
- Request reporting occurs on the primary blade, and every entry has the ID number of a slot on which the request has been processed.
- The full configuration is synchronized across all cluster members once an hour.
- The IP address enforcement feature and associated statistics are not available on VIPRION systems.

For more information about configuring the VIPRION chassis, refer to the ***Configuration Guide for the VIPRION® System***.

◆ Note

When a new primary cluster member is elected within Local Traffic Manager™, the Application Security Manager applies the full configuration of the new primary cluster member across all other cluster members. For more information on working with the Local Traffic Manager, refer to the ***Configuration Guide for BIG-IP® Local Traffic Manager™***.

Viewing cluster statistics

You can view statistics for all active blades running on the VIPRION chassis.

To view statistics for all blades

In the navigation pane, expand **Application Security** and click **Overview**. The Overview screen opens and displays statistics for the system including all blades running on the VIPRION chassis.

Viewing VIPRION cluster member synchronization status

The Application Security Manager displays the synchronization status for each cluster member in the VIPRION chassis in the context of security policies. Although each cluster member has its own Configuration utility, you can view the synchronization status only from the primary cluster member. The possible status for each blade is:

- **Up to date**
  The security policy for this cluster member is identical to that of the primary cluster member.
- **Waiting for reply**
  The security policies for this cluster member have not yet received the security policy update.
- **Loading**
  The system is currently applying policy changes to this cluster member to synchronize it with security policy changes made on the primary cluster member.
- **Error**
  The system was not successful in applying security policy changes from the primary cluster member. As a result, the active security policy on this cluster member is different from the active security policy on the primary member.

To view cluster member synchronization status

1. In the navigation pane, expand **Application Security** and click **Overview**.
   The Overview screen opens, and the system displays a summary of all configured web applications and security policies, and includes graphs with statistical information regarding traffic to the web applications.
2. Click **Synchronization Status**.
   The Synchronization Status screen opens, where you can review which slot is designated as the primary cluster member of the VIPRION system, and the security policy enforcement status of each secondary cluster member relative to the primary cluster member. The cluster member status changes if you perform the **Apply Policy** action or make any change that is immediately enforced, for example, install a UCS file, change a logging profile, and import or export a security policy.

Glossary

access violation

An access violation is a security policy violation that occurs when an HTTP request tries to gain access to an area of a web application, and some entity in the request does not comply with the security policy. See also *cookie violation*, *entity*, *input violation*, *length violation*, *negative security violation*, *RFC violation*, *security policy violation*.

Action Message Format (AMF)

Action Message Format (AMF) is a binary format that is loosely based on the Simple Object Access Protocol (SOAP). AMF is used primarily to exchange data between Adobe® Flash® applications and a database, by using the RPC (remote procedure call) protocol.

active security policy

The active security policy is the security policy whose criteria are determining the legitimacy of incoming requests for the web application. A web application can have only one active policy at a time.

application flow

See *flow*.

application security class

An application security class is the logical bridge, or link, between the local traffic components and the application security components of a BIG-IP system. You use the application security class to specify to which incoming HTTP traffic the system applies application security.

attack signature

An attack signature is a rule or pattern that identifies attacks or classes of attacks on a web application and its components. See also *attack signature set*, *system-supplied attack signatures*.

attack signature set

An attack signature set is a grouping of individual attack signatures. Rather than apply individual attack signatures to a security policy, you apply one or more attack signature sets. See also *attack signature*.

blocking actions

The blocking actions specify what the Security Enforcer does when a request does not comply with the active security policy. The blocking actions include the Learn flag, the Alarm flag, and the Block flag. When enabled, the Security Enforcer processes the requests according to the flags. See also *blocking mode*, *blocking policy*.

blocking mode

A security policy is in blocking mode when the enforcement mode is blocking, and one or more Block flags are enabled. In blocking mode, when a request triggers a violation, rather than forwarding the request to the corresponding web application, the Application Security Manager returns the blocking response page, which includes a Support ID, to the client. See also *enforcement mode*, *Support ID*, *transparent mode*.

blocking policy

The blocking policy specifies how the Security Enforcer processes a request (or response) that does not comply with the active security policy. The blocking policy is made up of the enforcement mode and the blocking actions (Learn, Alarm, and Block flags). See also *blocking mode*, *blocking actions*.

blocking response page

The blocking response page is the default response page that the Security Enforcer returns to a client when the client request, or the web server response, is blocked by the security policy.

buffer overflow

A buffer overflow occurs when an application attempts to store more data in a temporary storage area than is allowed. When data in a buffer exceeds the size of the buffer, adjacent buffers can overflow, corrupting the data already stored there. In a buffer overflow attack, an attacker can incorporate additional codes designed to trigger specific actions which could send new instructions to the attacked system in order to damage the user's files, change data, or disclose confidential information.

character set

A character set is a collection of alphabet and meta characters for a language. See also *meta character*.

cookie

A cookie is a message sent to a Web browser by a Web server, that the server can retrieve at a later time. The browser stores the message in a text file. Cookies are usually used to track a user’s actions when browsing a site.

cookie manipulation

Cookie manipulation is the process of altering or modifying cookie values on a client system’s web browser in order to exploit security issues within a web application. An attacker can manipulate cookie values on the client system to fraudulently authenticate themselves to a web site. See also *cookie*.

cookie violation

A cookie violation is a security policy violation that occurs when the cookie values in the HTTP request differ from those defined in the security policy. See also *access violation*, *entity*, *input violation*, *length violation*, *negative security violation*, *RFC violation*, *security policy violation*.

cross-site scripting

Cross-site scripting (XSS) is a type of exploit where information from one context, where it is not trusted, can be inserted into another context, where it is. For example, an attacker can insert malicious coding into a link that appears trustworthy, but when a user follows the link, the embedded code is submitted as a part of the client system's request, which could allow the attacker access to the client system.

Denial of Service

Denial of Service (DoS) is an attack technique on a network or web site that is designed to render the network or site useless by flooding it with excessive traffic. Processing the excess traffic can consume CPU cycles, memory usage, traffic bandwidth, and disk space, causing the system to become inaccessible to normal activity.

deployment scenarios

When you use the Deployment wizard, deployment scenarios represent several typical environments that use application security, to guide you through the configuration process.

Deployment wizard

The Deployment wizard automates the fundamental tasks required to initially build and deploy a security policy. See also *deployment scenarios*.

directory traversal

Directory traversal is an exploit that lets attackers access restricted directories and execute commands in areas beyond the normal web server directory. User access to web sites is typically restricted to the document root directory, or CGI root directory.

Dynamic content value (DCV) parameter

A DCV parameter is one for which the web application sets the value on the server side. See also *dynamic parameter*.

dynamic parameter

A dynamic parameter is a parameter whose set of accepted values can change, and usually depend on the user session. For example, within a banking web application, the account number parameter is a dynamic parameter, since each user has one or more unique account numbers. See also *static parameter*.

dynamic value

See *dynamic parameter*.

enforcement mode

The enforcement mode determines what actions the Security Enforcer takes when a request or response triggers a security policy violation. See also *blocking mode, transparent mode*.

entity

An entity is one of the many components of a web application. File types, URLs, parameters, headers, methods, and character sets are all examples of entities.

entry point

An entry point is a web page from which a user can access the corresponding web application.

evasion technique

Evasion techniques are coding methods for attacks that designed to avoid detection by attack signatures. See also *attack signature*.

false-positive alarm

False-positive alarms occur when the system blocks a request that is actually legitimate. false-positive alarms are also known as *false-positives*.

file type

A file type is a type of file used in the web application, usually referred to by its file extension. For example, JSP, ASP, GIF, and PNG are file types.

flow

Flow is the defined access path for a browser to get from one URL to another specific URL within a web application. Flow is also known as *application flow*.

flow parameter

Parameters that are defined within the context of an application flow are known as flow parameters. See also *global parameter, URL parameter*.

global parameter

Within the Application Security Manager configuration, global parameters are defined parameters that are not associated with a specific URL or a specific application flow. The Security Enforcer validates global parameters wherever they occur in the web application. See also *flow parameter, URL parameter*.

headers

See *HTTP headers*.

heuristics

Heuristics are the data collected and analyzed by algorithms in the Policy Builder. The Policy Builder uses the heuristics to make decisions regarding additions and updates to security policy entities. See also *entity*.

HTTP (HyperText Transfer Protocol)

HyperText Transfer Protocol (HTTP) is the protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted, and how a web browser requests data and how a web server responds.

HTTP class

See *application security class*.

HTTP headers

In an HTTP request, the HTTP headers specify the behavior and characteristics of the request.

HTTP method

In an HTTP request, the HTTP method (or simply, *method*) indicates the action that the client would like the server to perform for the requested resource. The most common methods are GET and POST.

input violation

An input violation is a security policy violation that occurs when an HTTP request includes a parameter or header that contains data or information that does not match, or comply with, the security policy. See also *access violation*, *cookie violation*, *entity*, *length violation*, *negative security violation*, *RFC violation*, *security policy violation*.

JavaScript

JavaScript™ is a scripting language that is used to create dynamic or interactive web page content.

learning process

The learning process is the process of making a security policy more accurate by verifying how the security policy complies with traffic requests. If the learning process finds discrepancies between the security policy and the traffic requests, it translates the discrepancies into a learning suggestion for modifying the security policy.

learning suggestion

When a request triggers a violation, and the Learn flag is enabled for that violation, the Learning Manager generates a learning suggestion. The learning suggestion contains information about what in the request caused the violation.

length violation

A length violation is a security policy violation that occurs when an HTTP request contains an entity that exceeds the length setting that is defined in the security policy. See also *access violation*, *cookie violation*, *entity*, *input violation*, *negative security violation*, *RFC violation*, *security policy violation*.

meta character

A meta character is a special character in a program or form field that can control or give information about other characters. They may have special meaning to programming languages, operating systems, or database queries. See also *character set*.

meta character injection

Meta character injection is an attack technique where an attacker sends meta characters as data input with the intent to manipulate a web application. See also *cross-site scripting*, *null injection*, *parameter tampering*, *SQL injection*.

method

See *HTTP method*.

negative security violation

A negative security violation is a security policy violation that occurs when an incoming request contains a string pattern that matches an attack signature in one of the security policy's attack signature sets, or when a response contains exposed user data, for example a credit card number. See also *access violation*, *cookie violation*, *entity*, *input violation*, *length violation*, *RFC violation*, *security policy violation*.

null injection

Null injection is an attack technique that bypasses sanity-checking filters by adding null-byte characters to a URL. If a user-input string contains a null character (0\), the web application on the site may stop processing the string at the null insertion point. This is a form of meta character injection. See also *meta character injection*, *parameter tampering*.

parameter and value pair

A parameter and value pair represents some element in a web application, usually a form field. When a web server receives a request that contains a parameter and value pair, the web server takes an action based on that input. Parameter and value pairs are found in the query string of a request URI. For example, the URI, **http://www.siterequest.com/login?username=joe&20password=12345**, contains two parameter and value pairs: **username=joe** and **password=12345**.

Note that parameter and value pairs are most often referred to simply as *parameters*. See also *parameter level*, *static parameter*, *dynamic content value (DCV) parameter*, *user-input parameter*, *XML parameter*.

parameter level

See *flow parameter*, *global parameter*, *URL parameter*.

parameter tampering

Parameter tampering is an attack technique in which the attacker tries to gain access to the web application by changing the parameter name and value pairs in a URL. This exploit is also referred to as *URL manipulation*. See also *URL manipulation*.

path traversal attacks

A path traversal attack is an HTTP attack technique that uses patterns like **../../** to get access to files not intended to be viewed above the WWW root, or in order to cross directories on the server.

profile

A profile is a BIG-IP system configuration tool that contains settings for defining the behavior of network traffic. See also *security profile*, *traffic profile*.

referrer

A referrer is a web page that can request other URLs. For example, an HTML page can request a GIF, JPG, or PNG file. The HTML page is a referrer; the image files are not.

regular expression

A regular expression (regexp or regex) is a sequence of characters that provides the user with a powerful, flexible, and efficient test processing tool.

remote procedure call (RPC) protocol

The remote procedure call (RPC) protocol allows a program on one computer to run a program on a server computer.

response scrubbing

The process of removing sensitive user information-such as credit card numbers, or social security numbers (U.S. only)-from a response to prevent exposure of the information to malicious users.

RFC violation

An RFC violation is a security policy violation that occurs because some part of a request or response does not comply with the HTTP protocol standards published in the HTTP RFC documents. The entire set of RFC documents is available at **http://www.ietf.org/rfc**. See also *access violation*, *cookie violation*, *entity*, *input violation*, *length violation*, *negative security violation*, *security policy violation*.

Secure Sockets Layer (SSL)

See *SSL (Secure Sockets Layer)*.

security policy

A security policy is a configuration of settings that secures traffic for a web application. It defines which traffic (such as which file types, URLs, parameters, and cookies) can access the application, and what happens to traffic that does not comply with the security policy. A security policy can also include anomaly detection, IP address enforcement, CSRF protection, mandatory headers, allowed methods, protection against web scraping, and many other security features. See also *security policy violation*.

security policy violation

A security policy violation indicates a breach of the rules specified in the security policy. A violation occurs when some aspect of a request or response does not comply with the security policy for a web application. See also *access violation*, *cookie violation*, *input violation*, *length violation*, *negative security violation*, *RFC violation*, *security policy*, *web application*.

security profile

A security profile is a system configuration tool in the Protocol Security Module that contains settings specific to securing network traffic. You associate security profiles with traffic profiles. See also *traffic profile*, *profile*.

session fixation

Session fixation is a technique that an attacker can use to force a different value to a user's session credential. See also *session ID*.

session hijacking

Session hijacking is the act of compromising a user's session. If an attacker hijacks a user's session, the attacker may appear to be the legitimate user to the web server. See also *session ID*.

session ID

A session ID is a string of data that identifies a user to a web server. This string can be contained in a cookie or in the URL. A session ID can track a user's session as he uses the web site.

Simple Object Access Protocol (SOAP)

SOAP (Simple Object Access Protocol) is the XML-based application protocol used to implement web services within a service-oriented architecture (SOA). SOAP is transported primarily using HTTP and middleware messaging systems, but can also be transported using other protocols such as SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol).

SQL injection

SQL injection is an attack technique used on database-driven web sites where an attacker runs unauthorized SQL commands by exploiting insecure code on a system to bypass the firewall in front of the SQL database. See also *parameter tampering*.

SSL (Secure Sockets Layer)

Secure Sockets Layer (SSL) is a standard protocol designed to provide an encrypted connection between two systems such as a web server and web browser. SSL uses two keys, a public key known to everyone, and a private key known to the recipient of the message.

staging

Staging is an interim test period which occurs when attack signatures or entities (such as a file types, URLs, or parameters) are first added to the security policy. When entities or attack signatures are in staging, you can test before enforcing them to see whether adding them to the security policy causes false positives or other problems to occur. The system provides learning suggestions for staged entities.

static parameter

A static parameter is a parameter in a request whose values are chosen from a known set of values, for example, the name of a country, a Yes/No form field, and so on. See also *dynamic parameter*.

static value

See *static parameter*.

Support ID

The Support ID identifies a request that triggers a security policy violation. When the enforcement mode is blocking, the system sends the blocking response page, which includes the Support ID, to the offending client. See also *blocking mode*, *blocking response page*, *enforcement mode*.

system-supplied attack signatures

System-supplied attack signatures are shipped as part of the Application Security Manager software. See also *attack signature, user-defined attack signature*.

target security policy

The target security policy is the security policy that the system updates whenever you accept a learning suggestion. See also *active security policy*.

tightening

Tightening is the process by which a security policy discovers the explicit file types, URLs, or parameters that match wildcard entities. See also *wildcard entity*.

traffic profile

A traffic profile is a BIG-IP system configuration tool that contains settings specific to the behavior of network traffic protocols, for example, HTTP, FTP, and SMTP. The terms *traffic protocol* and *profile* may be used interchangeably. See also *profile*, *security profile*.

transparent mode

When the enforcement mode for a security policy is transparent, the Security Enforcer forwards all requests to the web application, even if a request triggers a security policy violation. See also *blocking mode*, *enforcement mode*.

trusted traffic

Trusted traffic is traffic generated by a controlled group of users, those who are known not to be potential attackers. Example sources of trusted traffic are internal test groups or employees, or traffic generated by users on an internal LAN.

URI (Universal Resource Identifier)

The Universal Resource Identifier (URI) specifies the name of a URL in a request. For example, in this web address **http://www.siterequest.com/index.html**, the URI is **/index.html**.

URL (Universal Resource Locator)

A Universal Resource Locator (URL) is the standard method for specifying the location of a web page on the Internet.

URL manipulation

URL manipulation describes the process of changing the parameter name and value pairs of a web application. Also known as *parameter tampering*.

URL parameter

An URL parameter is a parameter that is defined and validated within the context of a URL. See also *flow parameter*, *global parameter*.

user-defined attack signature

A user-defined attack signature is an attack signature that a user writes and adds to the attack signatures pool. See also *attack signature*, *system-supplied attack signatures*.

user-input parameter

A user-input parameter requires users to enter or provide some sort of data. Comment, name, and phone number fields on an online form are all examples of user-input parameters.

violation

See *security policy violation*.

web application

A web application is an application delivered to users from a web server to a web client, such as a web browser, over a network. See also *web service*.

web object

See *URI (Universal Resource Identifier)*, *URL (Universal Resource Locator)*.

web object parameter

See *URL parameter*.

web service

A web service is a self-contained, self-describing, modular web application that can be published, located, and invoked across the Web. See also *web application*.

wildcard entity

A wildcard entity is a web application entity in the security policy that contains one or more shell-style wildcard characters in its name. You can use wildcard entities to represent file types, URLs, and parameters. See also *dynamic parameter*, *entity*, *file type*, *global parameter*, *URL (Universal Resource Locator)*, *URL parameter*, *user-input parameter*.

XML parameter

An XML parameter is a parameter whose value contains XML data.

Index

A

About tab 1-3
abuse of functionality attack 11-3
Accept as Legitimate (Loosen) rule 5-15, 5-17
access validation
  and brute force attack protection 7-11
access violations A-5
Action Message Format (AMF)
  configuring for URLs 6-27
Active icon 6-13
active security policy
  setting 4-4, 6-12
ActiveSync application-ready security policies B-3
actor, security header 12-8
Adobe Flash applications 6-27
Advanced settings, displaying by default 14-2
Alarm flag 6-43
Allow Empty Value setting
  configuring 10-20
  configuring for global parameter 10-3, 10-6, 10-9
Allow Repeated Occurrences setting 10-21
allow_all_cookies_at_entry_point parameter D-1
allowed file types
  defined 6-17
  properties of 6-17
allowed HTTP methods 6-40
allowed meta characters 10-15
allowed methods
  adding 6-40
  editing 6-40
allowed modified cookies
  defining 6-36
  deleting 6-38
  editing 6-37
  enforcing wildcards 9-20
  using wildcards 9-18
allowed response status codes, modifying 6-8
allowed URLs, creating 6-24
AMF requests
  and Content-Type header 6-27
  configuring security for 6-28
  determining 6-27
anomaly detection
  and VIPRION F-1
  configuring IP address enforcement 7-12
  detecting web scraping 7-13
  overview 7-1
  preventing brute force attacks 7-6, 7-7
  preventing DoS attacks 7-2, 7-3, 7-12, 7-14
anomaly statistics
  viewing 15-12
  viewing overview 15-2
anti-virus protection, configuring 14-3
AOL, and web scraping 7-14
application flow
  about 6-30
  and mandatory parameters 10-9
  and parameters 10-8
  See also flows.
application security class
  and web applications 4-6
  configuring 3-8
  creating 2-3, 3-2
  defined 2-3, 3-1
  disabling web applications 4-8
  naming 4-8
  processing HTTP requests 3-1
  redirecting action 3-8
  rewriting a URI 3-9
  sending to pool action 3-8
  using traffic classifiers 3-1, 3-3
Application Security setting 3-1
application-ready security policies
  about B-1
  and Deployment wizard B-1
  and PeopleSoft Portal 9 B-11
  for ActiveSync application B-3
  for Lotus Domino 6.5 application B-8
  for Oracle Applications 10g application B-9
  for Oracle Applications 11i application B-10
  for OWA Exchange 2003 application B-4
  for OWA Exchange 2007 application B-5
  for SAP NetWeaver application B-12
  for SharePoint 2003 application B-6
  for SharePoint 2007 application B-7
  for WhiteHat Sentinel B-13
  managing large file uploads B-14
ArcSight logs 14-9
ask.com, and web scraping 7-14
ASM cookie D-1
ASM cookie hijacking violation A-11
ASM_REQUEST_BLOCKING event 6-10
ASM_REQUEST_VIOLATION event 6-10
ASM_RESPONSE_VIOLATION event 6-10
assertions, in attack signatures C-14
attack mitigation, for DoS attacks 7-3
Attack signature detected violation 11-2, A-12
attack signature risk
  defined 11-7, 11-8
attack signature sets
  and blocking policy 11-20
  assigning to a security policy 11-13
  creating filter-based 11-14
  creating manual 11-15
  defined 11-2
  deleting 11-16
  editing 11-16
  including system-supplied 11-2

attack signature updates
and network access 11-10
and update failures 11-10
receiving email notification 11-12
viewing update activity 11-12
attack signatures
and blocking policy 11-2, 11-20
and custom filter options 11-7
and normalizing parameters C-4
and normalizing URIs C-4
and trusted traffic 11-23
assigning to parameters 10-15
configuring accuracy 11-8
creating for parameters C-14
creating user-defined C-1
defined 11-1
disabling 11-19, 11-23, C-14
enabling 11-23
enabling staging 11-21
enforcing after staging 11-24
escaping special characters C-13
for requests C-5
for responses C-5, C-14
staging 11-21, 11-23
tracking disabled D-2
updating automatically 11-11
updating considerations 11-10
updating manually 11-11
using Filter option 11-19
using XML format 11-28
viewing 11-18
viewing details 11-8
viewing revision number 11-9
viewing risk of 11-8
See also parameter attack signatures.
See also response signatures.
See also user-defined attack signatures.
attack signatures pool
about 11-1, 11-6
creating a custom filter 11-7
filtering view of 11-6
viewing 11-17
attack types 11-3, A-13
attacks
configuring DoS attack mitigation 7-3
detecting patterns 11-21
detecting possible 15-1
preventing brute force 7-6
preventing buffer overflow 6-6
attribute values, setting maximum length 12-18
attributes, specifying maximum number per element 12-18
audit tools 8-11
authentication
and attack signatures 11-3
configuring logon credentials 7-7
monitoring failures 7-6
restricting URLs 6-34
authorization attacks 11-3
automatic policy building
changing policy type 5-6
configuring advanced settings 5-4
configuring basic settings 5-2
modifying options 5-11
modifying rules 5-17
overview 5-1
restoring default values 5-20
stopping and starting 5-23
understanding rules 5-15
viewing status 5-21

B

backdoor attack 11-5
Basic settings, displaying by default 14-2
binary data type, configuring 10-16
binary export of requests 15-7
Block flag 6-43
blocked IP addresses
configuring IP Enforcer 7-12
releasing 15-14
viewing 15-13
blocked requests 6-46
blocking mode
and blocking response page 6-46
and support ID numbers 6-3
configuring 6-4, 6-42
defined 6-3
blocking policy
and attack signature staging 11-21
configuring 6-41, 6-43
configuring for evasion techniques 6-44
disabling 13-16
for attack signature sets 11-2, 11-20
setting blocking actions 6-43
blocking response page
and blocking mode 6-3
configuring 6-42
customizing 6-46
sending 6-43
bot activity, preventing 7-13
brute force attacks
and access validation 7-11
defined 11-3
mitigating 7-6
viewing reports 15-13

buffer overflow attacks
and length violations A-6
description 11-3
preventing 6-6, 6-7
buffer size, request D-3

C

CDATA, allowing in XML request 12-18
certificates
uploading for web services 12-6
character set
and language encoding 4-3
for parameters 10-29
for URLs 6-28
See also default character set.
charts
interpreting 15-10
sending via email 15-11
viewing 15-8
Charts Scheduler 15-11
Check AMF setting 6-23
Check Flows to this URL setting 6-22
Check Response setting 6-18
children, specifying maximum number per parent 12-18
classes
configuring application security 2-3, 3-2, 3-8
defined 3-1
disabling web applications 4-8
See also application security class.
close tag format, tolerating in XML requests 12-17
command execution attack 11-3
command injection attack 11-2
Common Event Format (CEF) 14-9
compliance
configuring HTTP 6-15
viewing PCI report 15-15
configuration tasks 2-1
Configuration utility
about 1-2
and online help 1-4
and the Welcome screen 1-4
overview 1-3
content rule option C-5
Content-Type header
and AMF requests 6-27
control characters
See non-printable characters.
Cookie not RFC-compliant violation A-3
cookie_digest_key parameter D-1
cookie_expiration_time_out parameter D-1
cookie_max_age parameter D-1
cookie_renewal_time_stamp parameter D-1
cookies
and Modified ASM cookie violation A-11
defining allowed modified 6-36
deleting allowed modified 6-38
editing allowed modified 6-37
enforcing wildcards for allowed modified 9-20
setting header length 6-7
using traffic classifier 3-7
using wildcards in headers 9-18
See also allowed modified cookies.
CPU usage 15-18
credit card numbers
and violations A-12
removing from responses 6-35
credit card type parameters 10-13
cross-site scripting (XSS) attacks 11-2, 11-3
cryptographic operations maximum D-2
CSRF attack detected violation 6-48, A-5
CSRF authentication expired 6-48
CSRF authentication expired violation A-5
CSRF session cookie A-5
custom filter, creating 15-17
custom patterns, sensitive data 6-35

D

Data Guard feature, configuring 6-35
data types
configuring alpha-numeric parameters 10-14
configuring binary parameters 10-16
configuring decimal parameters 10-17
configuring email parameters 10-17
configuring integer parameters 10-18
configuring phone parameters 10-19
DCV parameters
about 10-12
and dynamic names 10-27
and extracted items configuration 10-26
and extraction methods 10-26
and extraction properties 10-24
configuring 10-24
decimal data type, configuring 10-17
decryption, web services 12-5
default blocking response page 6-46
default character set
and language encoding 10-29
restoring 6-29
default sensitive parameter 10-31
defense configuration
configuring settings 12-17
defined 12-16
for XML profiles 12-16
defense level 12-16
defense level, protecting XML documents 12-17

denial-of-service attacks
defined 7-2, 11-3
mitigating 7-3
recognizing 7-2
deployment scenarios 2-5
Deployment wizard
about 2-5
and application-ready security policies B-1
and assigning attack signature sets 11-17
and configuring security policies 6-1
and deployment scenarios 2-5
depth modifier syntax C-9
detection criteria
for brute force attacks 7-9
for DoS attacks 7-5
detection evasion attack 11-3
detection interval 7-3, 7-6
digital signatures
implementing web services security 12-5
directory indexing attack 11-3
directory traversal 11-2
disallowed file types 6-16, 6-20
disallowed meta characters, configuring 10-15
disallowed URLs, configuring 6-26
distance modifier syntax C-10
document size, setting for XML 12-18
Document Type Definition (DTD) 12-17
DoS attacks
See denial-of-service attacks.
DoS Attacks reports, viewing 15-12
dynamic content value (DCV) parameters
See DCV parameters.
dynamic flows, configuring 6-32
dynamic mitigation 7-8
dynamic parameter names
about 10-12
and DCV parameters 10-27
and flow parameters 10-27
configuring 10-27
dynamic parameters
and Illegal parameter violation A-10
configuring 10-24
identifying 5-11
See also static parameters.
dynamic session IDs 6-8
dynamic session IDs in URLs, configuring 6-9

E

ecard_max_http_req_uri_len parameter D-1
ecard_regexp_decimal parameter D-1
ecard_regexp_email parameter D-2
ecard_regexp_phone parameter D-2
editing context area, described 8-2
elements, setting maximum number in XML document 12-18
email charts 15-11
email data type, configuring 10-17
email valid value D-2
email, configuring SMTP 14-14
empty values, allowing 10-20
encryption, web services 12-5
Enforce all URLS setting 6-36
Enforce Signatures button 11-24
enforcement mode
configuring 6-3, 6-42
defined 6-3
enforcement order
defined 9-8, 9-12, 9-16
setting for wildcard file type 9-8
setting for wildcard parameter 9-16
setting for wildcard URLs 9-12
enforcement, IP address 7-12
Enforcer statistics, viewing 15-13
enterprise applications
creating security policies for B-1
entities
adding to security policy 13-13
configuring the staging-tightening period 6-5
merging security policies 8-5
staging 13-11
staging and tightening 13-9
tightening 13-10
understanding wildcard 9-1
viewing ignored 13-18
entry point, application 6-22, 6-31
Evasion technique detected violation A-3
evasion techniques
configuring blocking properties 6-44
described 6-41
mitigating C-4
event severity levels, setting 14-11
exception patterns, sensitive data 6-36
expiration, login 6-34
Expired timestamp violation A-11
explicit file types 6-16
explicit URLs
configuring 6-24
described 6-21
export Requests List 15-7
export security policy 8-3
external references, allowing in XML requests 12-17
extractions
configuring DCV parameters 10-24
definition 6-23
viewing all 10-27
viewing for URLs 10-27

F

F5 Dev Central web site 3-3
failed login attempts 7-6, 7-10

Failure to convert character violation A-8
false positives
and accuracy 11-8
and attack signatures in staging 11-23
eliminating 13-1
file type properties, table of 6-17
file types
adding 6-16
and case-sensitivity 6-16
configuring allowed 6-16
creating allowed 6-18
creating wildcards 9-5
deleting wildcards 9-7
disallowing 6-20
modifying 6-19
modifying wildcard 9-6
removing from security policy 6-19
filter reports 15-17
filter-based signature sets 11-14
flow parameters
and Allow Empty Value option 10-20
and dynamic parameter names 10-27
and referrer URLs 10-8
configuring 10-8
configuring Is Mandatory Parameter setting 10-22
deleting 10-11
editing 10-10
flows
creating manually 6-31
definition 6-23, 6-30
viewing application 6-30
viewing for URLs 6-30
forceful browsing
definition 11-3
preventing with login URLs 6-33
FTP connections, setting maximum number D-2

G

general system events 14-12
general system options 14-1
Generic Detection Signatures set 11-17
GET method 6-40
global parameters
and Allow Empty Value option 10-20
and security level 10-2
creating 10-2
defined 10-2
deleting 10-4
editing 10-4
global security policy settings 10-15
Google, and web scraping 7-14
Grace Interval setting (web scraping) 7-14
GUI preferences 14-2

H

HEAD method 6-40
headercontent rule option C-6
headers
configuring mandatory 6-39
using traffic classifier 3-6
Help tab 1-3
help, online 1-4
hierarchy, viewing security policy 8-10
hijacking, session 11-5
history interval 7-3, 7-6
hosts traffic classifier 3-3
HTTP class
See application security class.
HTTP flood attack
See denial-of-service attacks.
HTTP methods 6-40
HTTP parameter pollution 10-21, A-9
HTTP parser attack 11-3
HTTP protocol
and application-ready security policies B-1
HTTP protocol compliance
configuring 6-15
validating requests 6-14
HTTP protocol compliance failed violation 6-14, A-4
HTTP request smuggling attack 11-4
HTTP response splitting 11-4
HTTP security profile
converting to security policy E-1
HTTP-GET attack
See denial-of-service attacks.
HTTPS protocol
and application-ready security policies B-1
human activity 7-13
human-readable security policy 8-3

I

ICAP server, configuring 14-3
ICSA-certified 1-1
ignored entities list
for web application 13-18
removing items from 13-18
Ignored Entities screen 13-1
Ignored IP Addresses screen 13-1
ignored IP addresses, creating 13-19
Illegal attachment in SOAP message violation A-8
Illegal cookie length violation A-6
Illegal dynamic parameter value violation A-8
Illegal empty parameter value violation 10-20, A-8
Illegal entry point violation A-5
Illegal File Type violation 6-20
Illegal file type violation A-5
Illegal flow to URL violation A-5
Illegal header length violation A-6
Illegal HTTP status in response violation 6-8, A-5

Illegal meta character in header violation A-8
Illegal meta character in parameter value violation A-8
Illegal meta character in parameter violation A-5
Illegal meta character in URL violation A-5
Illegal method violation A-5
Illegal number of mandatory parameters violation A-8
Illegal parameter data type violation A-9
Illegal parameter numeric value violation A-9
Illegal parameter value length violation A-9
Illegal parameter violation A-8, A-10
Illegal POST data length violation A-6
Illegal query string length violation A-6
Illegal Query-String or POST Data violation A-9
Illegal repeated parameter name violation 10-21, A-9
Illegal request length violation A-7
Illegal session ID in URL violation 6-8, A-5
Illegal static parameter value violation A-9
Illegal URL length violation A-7
Illegal URL violation 6-26, A-6
information leakage attack 11-4
Information leakage detected violation 6-35, A-12
Injection attempt 11-4
input violations
  summary of A-8
instructions, allowing in XML request 12-18
integer data type
  configuring 10-18
internal parameters
  described D-1
  restoring default settings D-5
  viewing D-4
IP address enforcement 7-12
IP address whitelist
  for DoS attacks 7-6
  for web scraping 7-14
IP addresses
  configuring trusted 5-19
  creating list to ignore 13-19
  deleting ignored 13-19
  releasing blocked 7-12, 15-14
  viewing blocked 15-13
  viewing top requesting 15-2
IP Enforcer
  releasing blocked IP addresses 15-14
IP Enforcer statistics, viewing 15-13
IP Enforcer, configuring enforcement 7-12
iRule events, activating 6-10
iRule, definition 6-10
Is Mandatory Parameter setting 10-9, 10-22

K

keyword modifiers
  for rule options C-2
  See also user-defined attack signatures.

L

language encoding
  and default character set 10-29
  setting for web application 4-3
  supporting double-byte 4-3
  supporting single-byte 4-3
latency mitigation 7-3, 7-4
LDAP injection attack 11-4
Learn flag
  about 6-43
  enabling learning suggestions 13-2
Learning Manager 13-1
learning process
  and length violations A-6
  overview 13-1
learning suggestions
  accepting 13-7
  and tightening 9-2, 13-10
  clearing 13-8
  displaying 13-1
  ignoring IP addresses 13-19
  interpreting 13-15
  processing 13-7
  rejecting 13-18
  viewing related requests 13-3
length violations A-6
local logging 14-5
Local Traffic Manager
  configuring HTTP class profiles 3-1
  integrating with 1-1
local traffic pool 2-1
local traffic virtual server
  See virtual server.
location directive 12-4
log files 14-12
  viewing the policy log 5-24, 8-9
logging profiles
  about 14-5
  and storage format 14-6
  configuring for a reporting server 14-8
  configuring for ArcSight logs 14-9
  configuring local storage 14-5
  configuring remote storage 14-6
  filtering logs 14-10
  setting for a web application 4-4
login attempts 7-6, 7-10
login page configuration 6-33
login page response 6-46
login page settings 6-34
Login URL bypassed violation 6-33, A-6
Login URL expired violation 6-33, A-6
login URLs, configuring 6-33
login violations 6-46
logout URLs 6-34
LogSignatures parameter D-2
long_request_buffer_size parameter D-2

Lotus Domino 6.5 security policy B-8

M

Main tab, about 1-3
Malformed XML data violation A-9
mandatory headers 6-39
Mandatory HTTP header is missing violation 6-39, A-4
mandatory parameters 10-9
manual signature sets, creating 11-15
Mask Data option 6-35, 6-36
masked sensitive XML data 12-19
masking process for sensitive data 6-36
max_concurrent_long_request parameter D-2
max_filtered_html_length parameter D-2
MaxFtpSessions parameter D-2
maximum HTTP header length 6-6
Maximum login attempts are exceeded violation A-9
maximum memory size D-3
MaximumCryptographicOperations parameter D-2
MaxJobs parameter D-2
MaxSmtpSessions parameter D-2
MaxViolationEntries parameter D-2
memory size, setting maximum D-3
merge mechanism 8-5
meta characters
  and parameter values 10-29
  configuring 10-15
  for user-input parameters 10-14
methods
  adding allowed 6-40
  using default allowed HTTP 6-40
Microsoft ActiveSync
  creating security policy for B-3
Microsoft Outlook Web Access
  and security policy for B-4, B-5
Microsoft SharePoint 2003
  creating security policy for B-6
Microsoft SharePoint 2007
  creating security policy for B-7
Migration wizard E-1
mitigation, for DoS attacks 7-3
Modified ASM cookie violation A-11
Modified domain cookie(s) violation 9-18, A-11
Modified icon
  and activating a security policy 6-13
monitoring tools
  about 2-6
  See also reporting tools.
MSN, and web scraping 7-14

N

names, setting maximum length 12-18
names, tolerating numeric in XML 12-17
namespace mappings, for XML security 12-10
namespaces
  setting maximum declarations 12-18
  specifying maximum length 12-18
navigation parameters, configuring 10-32
negative security violations
  about A-12
  types of A-12
no extension file types 6-16
no_ext file type 6-16
nocase modifier syntax C-8
Non-browser client 11-4
Non-existent URL violation
  See Illegal URL violation.
non-printable characters, displaying 13-3
norm modifier syntax C-12
not character
  using in attack signatures C-2
Null in multi-part parameter value violation A-9

O

objonly modifier syntax C-12
offset modifier syntax C-9
online help 1-4
option clusters C-14
options, general system 14-1
Oracle Applications 10g security policy, configuring B-9
Oracle Applications 11i security policy, configuring B-10
Overview screen 15-2
OverviewEnabled parameter D-2
OWA Exchange 2003 security policy, configuring B-4
OWA Exchange 2007 security policy, configuring B-5

P

page flood attack
  See denial-of-service attacks.
paramcontent rule option
  about C-6
  using norm modifier C-12
parameter attack signatures
  about 11-2
  developing user-defined C-14
parameter name character set 10-30
parameter pollution 10-21, A-9
parameter tampering 11-4
parameter types 10-12
parameter value character set 10-29
Parameter value does not comply with regular expression violation A-9
parameter values
  and allowed meta characters 10-15
  and disallowed meta characters 10-15
  and meta characters 10-29
  ignoring 10-12

parameters
allowing empty value 10-20
allowing repeated occurrences of flow 10-9
allowing repeated occurrences of global 10-3
allowing repeated occurrences of URL 10-6
allowing repeated occurrences of wildcard 9-14
and application flows 10-8
and Is Mandatory Parameter setting 10-22
and Security Enforcer 10-1, 10-12
and XML profiles 12-22
assigning attack signatures 10-15
configuring navigation 10-32
configuring user-input 10-14
creating flow parameters 10-8
creating global parameters 10-2
creating URL parameters 10-5
deleting wildcard 9-15
identifying dynamic 5-11
modifying wildcard 9-15
viewing character sets 10-29
password attacks 7-6
password sensitive parameter 10-31
path traversal attack 11-4
Payment Card Industry (PCI) standards 15-15
PCI Compliance report 15-15
PCI-DSS 1.2 15-15
pcre action modifiers C-7
PCRE regular expressions
and Data Guard feature 6-35
pcre rule option
about C-2, C-6
and response rules C-14
and scopes C-3
escaping characters C-14
using C-6
using examples C-15
using modifiers C-7
PDF export of requests 15-7
penetration testing 13-19
PeopleSoft Portal 9
and application-ready security policies B-11
phone data type
configuring 10-19
phone number valid value D-2
Policy Builder
stopping and starting 5-23
policy log 5-24, 8-9
Policy Recycle Bin 8-7
policy type, changing 5-6
pool, defining local traffic 2-2
positive security model 1-2
POST data length 6-18
POST data, and XML profile 12-20
POST method 6-40
predefined filter 15-17
predictable resource location attack 11-4
preferences, configuring system and GUI 14-2
product documentation, finding 1-4
profile, logging 4-4
Protocol Security Module, migrating from E-1
ProtocolIndication parameter D-3
PRXRateLimit parameter D-3

Q

query strings
and dynamic sessions in URLs 6-9

R

RAM cache, and web scraping 7-13
Rapid Deployment security policy
about B-2
rate limiting
configuring for brute force 7-10
configuring for DoS attacks 7-4
Reconfigure button 4-5
records per screen, configuring 14-2
recycle bin, security policy 8-7
redirect action
in application security class 3-8
redundant configuration, recommending sync 14-2
reference rule option C-8
referrer URLs
and dynamic flows 6-32
and flow parameters 10-8
configuring for flow parameters 10-9
configuring in flows 6-31
RegExp Validator 14-13
regular expressions 3-3
in user-input parameters 10-14
using in internal parameters D-2
regular expressions, validating 14-13
release notes, finding 1-4
Remote file include 11-4
remote logging
configuring 14-6
reporting tools
about 2-6, 15-1
reports
filtering 15-17
viewing brute force attacks 15-13
viewing DoS attacks 15-12
viewing graphical 15-8
viewing PCI compliance 15-15
viewing web scraping 15-14
Request length exceeds defined buffer size violation A-6
disabling B-14
request signatures
about 11-2
See also attack signatures.
request_buffer_size parameter D-3

requests
clearing from the Requests List 15-7
configuring default number displayed 14-2
exporting 15-7
filtering by attack type A-13
logging 13-18
setting maximum number D-2
setting maximum request length D-2
setting the log level 4-4
viewing a full request 15-5
viewing details and violations 15-5
viewing reports 15-4
Requests List 15-4
Requests screen 15-4
response attack signatures
syntax considerations for user-defined C-14
response page 6-42
response scrubbing
configuring 6-35
response signatures 11-2
response status codes, configuring allowed 6-8
ResponseBufferSize parameter D-3
responses, setting maximum size D-2
Restore Defaults button 5-20
rewrite URI
in application security class 3-9
RFC compliance with HTTP 6-14
RFC documents A-3
RFC violations A-3
role, security header 12-8
RPC protocol 6-27
rule options
and scopes C-3
and syntax and usage C-5
combining C-14
defined C-1
escaping special characters C-13
for attack signatures C-4
using content C-5
using depth modifier C-9
using distance modifier C-10
using headercontent C-6
using keyword modifiers C-2
using nocase modifier C-8
using norm modifier C-12
using objonly modifier C-12
using offset modifier C-9
using paramcontent C-6
using pcre C-6
using the not character C-2
using uricontent C-5
using within modifier C-11
writing response rules C-14
rules, automatic policy building 5-15
RWLightThreads parameter D-3
RWThreads parameter D-3

S

Safe Interval setting (web scraping) 7-14
SAP NetWeaver application-ready security policies, described B-12
scanner IP address, ignoring 13-19
schema files, validating 12-3
schema links 12-4
and verifying 12-3, 12-23
schemaLocation directive 12-4
scopes
and pcre rule option C-3
for attack signature rules C-3
Security email distribution list 11-12
Security Enforcer
and parameters 10-12
disabling attack signatures 11-21
enforcing explicit entities 9-4
enforcing parameters 10-1
enforcing wildcard entities 9-4, 9-5
enforcing wildcard parameters 9-16
enforcing wildcard URLs 9-12
protecting XML data 12-20
verifying parameters 10-1
security events, filtering by web application group 4-6
security headers
processing requests 12-8
security policy
and access violations A-5
and DCV parameters 10-25
and enforcement mode 6-3
and length violations A-6
and negative security violations A-12
and sensitive parameters 10-31
assigning attack signature sets 11-13
configuring blocking mode 6-46
configuring properties 6-1
copying 8-3
creating a backup 8-3
creating automatically 5-2
defined 6-1
deleting permanently 8-7
enabling dynamic session IDs in URLs 6-8
enforcing parameters 10-2
exporting 8-3
finding version number 8-8
fine-tuning 13-1
implementing 2-1
importing 8-4
maintaining 8-1
merging two policies 8-5
migrating HTTP security profile E-1
monitoring 2-6
naming convention 8-4
removing from the configuration 8-6
removing URLs 6-25
resolving errors 8-11

restoring 8-7
restoring archived version 8-8
setting active 4-4, 6-1, 6-12
updating 13-2
using application-ready security policies B-1
using learning suggestions 13-7
viewing 8-11
viewing all changes 8-9
viewing automatic changes 5-24
security policy archives 8-8
security policy audit tools 8-11
security policy elements
and policy types 5-7
modifying 5-9
security policy properties
and maximum HTTP header length 6-6
configuring maximum cookie header length 6-7
Security Policy Recycle Bin 8-7
security policy tree view 8-10
security policy versions 8-8
security policy violations
about A-1
detecting legitimate 13-2
overview 6-41
tracking trends 15-1
viewing details 15-5
See also violations.
security reports
overview 15-1
viewing graphical charts 15-8
See also reports.
send to pool action
in application security class 3-8
sensitive data
managing 10-31
masking 6-35
masking in responses 6-36
masking XML 12-19
sensitive parameters
configuring in flow parameters 10-9
configuring in global parameters 10-3
configuring in URL parameters 10-6
deleting 10-31
editing 10-31
in web applications 10-31
masking in XML documents 12-19
Sensitive Parameters property
configuring 10-31
server-side code injection attack 11-4
session hijacking 11-5
session IDs, configuring dynamic 6-8
session token 11-5
session-based mitigation 7-8
sessions, setting maximum number D-2
severity level
for violations 14-11
SharePoint 2003 application-ready security policies B-6
SharePoint 2007 application-ready security policies B-7
signature sets
See attack signature sets.
SMTP configuration 14-14, 15-11
SMTP connections, setting maximum number D-2
SMTP mailer 14-14
SOAP messages 12-5
SOAP method not allowed violation 12-13, A-9
SOAP methods
validating 12-13
SOAP web services
configuring digital signatures 12-5
configuring security 12-3
social security numbers
removing from responses 6-35
special characters
using in attack signature rules C-13
spyware attack 11-5
SQL injection attack 11-5
Stabilize (Tighten) rule 5-15, 5-17
staging
and wildcard entities 9-2
configuring for attack signatures 11-21
configuring for parameters 10-2
configuring for URLs 6-21
configuring in file types 6-17
definition 11-21, 13-11
reviewing status 13-12
understanding 13-11
viewing summary of entities 13-9
staging period
and blocking policy 11-20
for attack signatures 6-6, 11-21
staging-tightening period, configuring 6-5, 6-6
Staging-Tightening screen 13-1
static content value parameters
See static parameters.
static parameters
about 10-12
configuring 10-13
See also dynamic parameters.
statistics
viewing anomaly 15-12
viewing application security overview 15-2
viewing IP Enforcer 15-13
viewing web scraping 15-14
status codes
configuring response 6-8
status, viewing automatic policy building 5-21
storage filter
configuring for logging profiles 14-10
storage format
for logging profiles 14-6

support ID numbers
and blocking mode 6-3
for security policy violations 13-4
in response pages 6-46
synchronization status, VIPRION F-2
syslog server
configuring remote logging 14-6
logging configuration changes 14-2
selecting facility 14-7
setting severity levels for violations 14-11
system messages, viewing 1-3
system options 14-1
system preferences, configuring 14-2
system resources
and logging profiles 14-5
managing 15-18
system-supplied attack signature sets 11-13
system-supplied attack signatures 11-1

T

Tcl expressions
rewriting URIs 3-9
using 3-3, 3-8
Technical Support web site 1-4
templates
using application-ready security policies B-1
threads, setting maximum number D-3
tightening
and creating wildcard file types 9-5, 9-9
and creating wildcard URLs 9-13
and learning suggestions 9-2, 13-10
and wildcard entities 9-2
configuring for allowed modified cookies 9-18
configuring for parameters 10-3
configuring for URLs 6-22
configuring in file types 6-17
reviewing status 13-12
understanding 13-10
viewing summary of entities 13-9
tightening period, configuring 6-5
tooltip settings, configuring 14-2
total_umu_max_size parameter D-3
total_xml_memory parameter D-3
Track Site Changes rule 5-15, 5-18
traffic classifiers
applying 3-3
for cookies 3-7
for headers 3-6
for hosts 3-3
for URI paths 3-5
in application security classes 3-1, 3-3
Traffic Learning screen 13-1
processing learning suggestions 13-7
traffic summary 15-2
transaction rate detection interval 7-2
transaction rate history interval 7-2
transparent mode
configuring 6-4, 6-42
defined 6-3
tree view of security policy 8-10
Trigger ASM iRule event check box 6-10
Trojan horse attack 11-5
Trust XFF Header check box 6-11
trusted IP addresses
configuring 5-19
trusted traffic
and attack signatures 11-23
trusted XFF headers, configuring 6-11

U

ultimateReceiver role 12-8, 12-10
UNNAMED parameter 10-2
upgrading software
and exporting security policies 8-3
URI length D-1
URI paths traffic classifier 3-5
uricontent rule option
about C-5
using objonly modifier C-12
URL parameters
and Allow Empty Value option 10-20
defining 10-5
editing 10-7
URLs
and application flow 6-30
and character sets 6-28
associating XML profiles 12-20
authenticating at logon 6-34
configuring disallowed 6-26
configuring dynamic flows 6-32
configuring explicit 6-24
configuring login 6-33
creating wildcards 9-9
defined 6-21
defining parameters for 10-5
deleting wildcards 9-11
enforcing Data Guard protection 6-36
modifying wildcards 9-11
removing from security policy 6-25
viewing extractions for 10-27
viewing properties of 6-25
viewing top requested 15-2
user activity
and application security 14-12
logging actions 14-12
user data
removing from responses 6-35
user interface preferences, configuring 14-2
user management 14-4

user roles
about 14-4
user-defined attack signatures
about 11-1
and failed attack signature updates 11-10
creating 11-26, C-1
deleting 11-27
exporting 11-29
importing 11-28
managing 11-25
modifying 11-27
using rule options C-1
See also attack signatures.
user-defined attack signatures syntax
See rule options.
user-input parameters
about 10-12
and alpha-numeric data type 10-14
and attack signatures 11-1
and binary data type 10-16
and character set 10-29
and configuring parameter characteristics 10-14
and decimal data type 10-17
and input violations A-8
and integer data type 10-18
and phone data type 10-19
configuring email data type 10-17
using meta characters in 10-14
using regular expressions 10-14
user-input value parameters
See user-input parameters.

V

verifying schema links 12-3, 12-23
version number, for security policy 8-8
View Full Request Information screen 13-4, 13-5
Viewing the list of extractions 10-27
violations
Attack signature detected violation A-12
clearing 13-17
Cookie not RFC-compliant A-3
disabling 13-16
Evasion technique detected A-3
Failure to convert character A-8
HTTP protocol compliance failed A-4
Illegal attachment in SOAP message A-8
Illegal cookie length A-6
Illegal dynamic parameter value A-8
Illegal entry point A-5
Illegal file type A-5
Illegal flow to URL A-5
Illegal header length A-6
Illegal HTTP status in response A-5
Illegal meta character in header A-8
Illegal meta character in header parameter value A-8
Illegal meta character in parameter A-5
Illegal meta character in URL A-5
Illegal method A-5
Illegal POST data length A-6
Illegal query string length A-6
Illegal session ID in URL A-5
Illegal URL A-6
Illegal URL length A-7
Information leakage detected violation A-12
Login URL bypassed A-6
Login URL expired A-6
Mandatory HTTP header is missing A-4
Request length exceeds defined buffer size A-6
requiring user interpretation 13-15
setting maximum number D-2
setting severity level 14-11
SOAP method not allowed A-9
viewing details 15-5
Web scraping detected A-9
XML data does not comply with format settings A-10
XML data does not comply with schema or WSDL document A-10
See also security policy violations.
VIPRION
and Application Security Manager F-1
and configuration F-1
and request reporting F-1
and synchronization F-1
overview of ASM running on F-1
viewing blade synchronization status F-2
virtual server
and application security class 3-1, 3-8
and iRule events 6-10
defining 2-4
Virus Detected violation 14-3
Virus detected violation A-12
virus_header_name parameter D-3
vulnerability scan attack 11-5

W

Web Accelerator cache, and web scraping 7-13
web application group
creating 4-7
defined 4-6
deleting 4-7
web application language
configuring 4-3
web application properties 4-3
web applications
and access violations A-5
and application security classes 4-6
and length violations A-6

and logging profiles 14-5
and negative security violations A-12
and sensitive parameters 10-31
configuring local logging 14-5
configuring remote logging 14-6
creating a default 2-3, 3-1
defined 4-1
defining parameters 10-1
deleting configurations 4-5
disabling 4-8
reconfiguring 4-5
setting active security policy 4-4, 6-12
setting language encoding 4-3
tightening security 10-1
viewing all 4-1
viewing disabled 4-8
viewing ignored entities 13-18
viewing requests for 15-4
web robots 7-13
web scraping
configuring detection 7-13
viewing reports 15-14
Web scraping detected violation 7-13, A-9
web services applications
configuring security policy 12-3
protecting 12-20
web services security
configuring 12-7
configuring blocking properties 6-45
handling encryption of data 12-1
implementing 12-5
writing XPath queries 12-12
Web Services Security failure violation 6-45, A-10
Welcome screen 1-4
white space, tolerating leading 12-17
WhiteHat Sentinel Baseline application-ready
security policy B-13
whitelist
for DoS attack mitigation 7-6
for web scraping 7-14
wildcard cookie headers 9-18
wildcard cookies
enforcing allowed modified 9-20
wildcard entities
about 9-1
and explicit entity matches 9-4
and wildcard entity matches 9-4
staging 9-3
staging and tightening 9-2
tightening 9-2
wildcard file types
and tightening 9-5, 9-9
creating 9-5
deleting 9-7
described 6-16
modifying 9-6
setting enforcement order 9-8
staging 9-3
wildcard parameters
about 9-13
creating 9-13
deleting 9-15
modifying 9-15
setting enforcement order 9-16
staging 9-3
wildcard syntax 9-1
wildcard URLs
and protecting web services applications 12-20
and tightening 9-13
creating 9-9
deleting 9-11
described 6-21
modifying 9-11
setting enforcement order 9-12
staging 9-3
within modifier syntax C-11
worms, protecting against 3-3
Write all changes to Syslog check box 14-2
Wrong message key violation
See ASM cookie hijacking violation.
WSDL documents
and valid SOAP methods 12-13
validating 12-3

X

XFF headers, configuring 6-11
X-Forwarded-For headers, configuring 6-11
XML data does not comply with format settings
violation A-10
XML data does not comply with schema
or WSDL document violation A-10
XML data, masking sensitive 12-19
XML file format
saving security policy 8-3
using for attack signatures 11-28
XML parameters
configuring 10-23
defined 10-13
XML parser attack 11-5
XML parser, setting maximum memory D-3
XML profiles
and defense configuration 12-16
associating with parameters 10-23, 12-22
associating with URLs 12-20
defined 12-3
deleting 12-24
validating schema files 12-3
validating WSDL files 12-3

XML security
    configuring for web services 12-3
    configuring for XML content 12-14
    encrypting SOAP messages 12-5
    overview 12-1
    verifying and signing SOAP messages 12-5
XML signatures
    implementing web services security 12-5
XPath queries, writing 12-12
XSS attacks 11-3

Y

Yahoo, and web scraping 7-14